The subdivision of the current probability interval would ideally require a multiplication of the interval by the probability estimate for the LPS. Because this subdivision is done approximately, it is possible for the LPS sub-interval to be larger than the MPS sub-interval. When that happens a "conditional exchange" interchanges the assignment of the sub-intervals such that the MPS is given the larger sub-interval.

Since the encoding procedure involves addition of binary fractions rather than concatenation of integer code words, the more probable binary decisions can sometimes be coded at a cost of much less than one bit per decision.

D.1.1.2 Conditioning of probability estimates

An adaptive binary arithmetic coder requires a statistical model – a model for selecting conditional probability estimates to be used in the coding of each binary decision. When a given binary decision probability estimate is dependent on a particular feature or features (the context) already coded, it is "conditioned" on that feature. The conditioning of probability estimates on previously coded decisions must be identical in encoder and decoder, and therefore can use only information known to both.

Each conditional probability estimate required by the statistical model is kept in a separate storage location or "bin" identified by a unique context-index S. The arithmetic coder is adaptive, which means that the probability estimates at each context-index are developed and maintained by the arithmetic coding system on the basis of prior coding decisions for that context-index.

D.1.2 Encoding conventions and approximations

The encoding procedures use fixed precision integer arithmetic and an integer representation of fractional values in which X'8000' can be regarded as the decimal value 0.75. The probability interval, A, is kept in the integer range X'8000' $\leq$ A < X'10000' by doubling it whenever its integer value falls below X'8000'. This is equivalent to keeping A in the decimal range $0.75 \leq A < 1.5$. This doubling procedure is called renormalization.

The code register, C, contains the trailing bits of the bit stream. C is also doubled each time A is doubled. Periodically – to keep C from overflowing – a byte of data is removed from the high order bits of the C-register and placed in the entropy-coded segment.

Carry-over into the entropy-coded segment is limited by delaying X'FF' output bytes until the carry-over is resolved. Zero bytes are stuffed after each X'FF' byte in the entropy-coded segment in order to avoid the accidental generation of markers in the entropy-coded segment.

Keeping A in the range $0.75 \leq A < 1.5$ allows a simple arithmetic approximation to be used in the probability interval subdivision. Normally, if the current estimate of the LPS probability for context-index S is Qe(S), precise calculation of the sub-intervals would require:

$Qe(S) \times A$	Probability sub-interval for the LPS;
$A - (Qe(S) \times A)$	Probability sub-interval for the MPS.

Because the decimal value of A is of order unity, these can be approximated by

Qe(S)	Probability sub-interval for the LPS;
A – Qe(S)	Probability sub-interval for the MPS.

Whenever the LPS is coded, the value of A – Qe(S) is added to the code register and the probability interval is reduced to Qe(S). Whenever the MPS is coded, the code register is left unchanged and the interval is reduced to A – Qe(S). The precision range required for A is then restored, if necessary, by renormalization of both A and C.

With the procedure described above, the approximations in the probability interval subdivision process can sometimes make the LPS sub-interval larger than the MPS sub-interval. If, for example, the value of Qe(S) is 0.5 and A is at the minimum allowed value of 0.75, the approximate scaling gives one-third of the probability interval to the MPS and two-thirds to the LPS. To avoid this size inversion, conditional exchange is used. The probability interval is subdivided using the simple approximation, but the MPS and LPS sub-interval assignments are exchanged whenever the LPS sub-interval is larger than the MPS sub-interval. This MPS/LPS conditional exchange can only occur when a renormalization will be needed.

Each binary decision uses a context. A context is the set of prior coding decisions which determine the context-index, S, identifying the probability estimate used in coding the decision.

Whenever a renormalization occurs, a probability estimation procedure is invoked which determines a new probability estimate for the context currently being coded. No explicit symbol counts are needed for the estimation. The relative probabilities of renormalization after coding of LPS and MPS provide, by means of a table-based probability estimation state machine, a direct estimate of the probabilities.

D.1.3 Encoder code register conventions

The flow charts in this annex assume the register structures for the encoder as shown in Table D.2.

Table D.2 – Encoder register connections

	MSB			LSB
C-register	0000cbbb,	bbbbbsss,	xxxxxxxx,	xxxxxxxx
A-register	00000000,	00000000,	aaaaaaaa,	aaaaaaaa

The "a" bits are the fractional bits in the A-register (the current probability interval value) and the "x" bits are the fractional bits in the code register. The "s" bits are optional spacer bits which provide useful constraints on carry-over, and the "b" bits indicate the bit positions from which the completed bytes of data are removed from the C-register. The "c" bit is a carry bit. Except at the time of initialization, bit 15 of the A-register is always set and bit 16 is always clear (the LSB is bit 0).

These register conventions illustrate one possible implementation. However, any register conventions which allow resolution of carry-over in the encoder and which produce the same entropy-coded segment may be used. The handling of carry-over and the byte stuffing following X'FF' will be described in a later part of this annex.

D.1.4 Code_1(S) and Code_0(S) procedures

When a given binary decision is coded, one of two possibilities occurs – either a 1-decision or a 0-decision is coded. Code_1(S) and Code_0(S) are shown in Figures D.1 and D.2. The Code_1(S) and Code_0(S) procedures use probability estimates with a context-index S. The context-index S is determined by the statistical model and is, in general, a function of the previous coding decisions; each value of S identifies a particular conditional probability estimate which is used in encoding the binary decision.

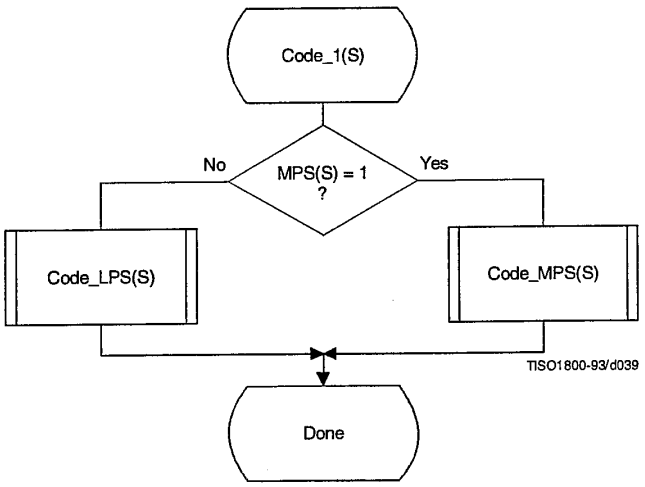


Figure D.1 – Code_1(S) procedure

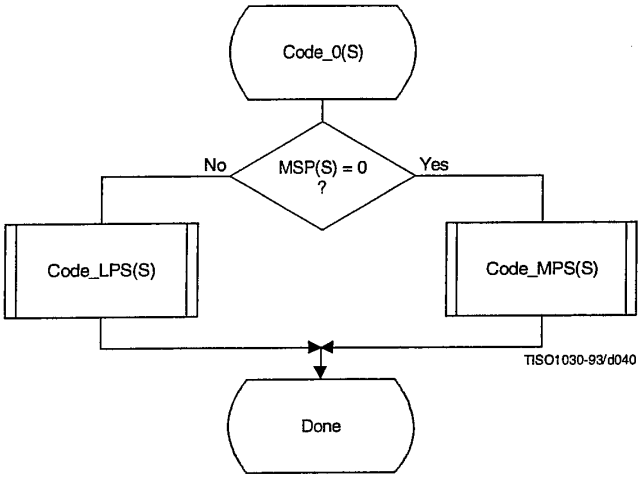


Figure D.2 – Code_0(S) procedure

The context-index S selects a storage location which contains Index(S), an index to the tables which make up the probability estimation state machine. When coding a binary decision, the symbol being coded is either the more probable symbol or the less probable symbol. Therefore, additional information is stored at each context-index identifying the sense of the more probable symbol, MPS(S).

For simplicity, the flow charts in this subclause assume that the context storage for each context-index S has an additional storage field for Qe(S) containing the value of Qe(Index(S)). If only the value of Index(S) and MPS(S) are stored, all references to Qe(S) should be replaced by Qe(Index(S)).

The Code_LPS(S) procedure normally consists of the addition of the MPS sub-interval A – Qe(S) to the bit stream and a scaling of the interval to the sub-interval, Qe(S). It is always followed by the procedures for obtaining a new LPS probability estimate (Estimate_Qe(S)_after_LPS) and renormalization (Renorm_e) (see Figure D.3).

However, in the event that the LPS sub-interval is larger than the MPS sub-interval, the conditional MPS/LPS exchange occurs and the MPS sub-interval is coded.

The Code_MPS(S) procedure normally reduces the size of the probability interval to the MPS sub-interval. However, if the LPS sub-interval is larger than the MPS sub-interval, the conditional exchange occurs and the LPS sub-interval is coded instead. Note that conditional exchange cannot occur unless the procedures for obtaining a new LPS probability estimate (Estimate_Qe(S)_after_MPS) and renormalization (Renorm_e) are required after the coding of the symbol (see Figure D.4).

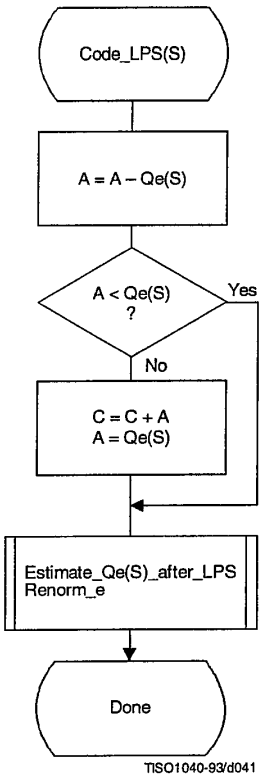


Figure D.3 – Code_LPS(S) procedure with conditional MPS/LPS exchange

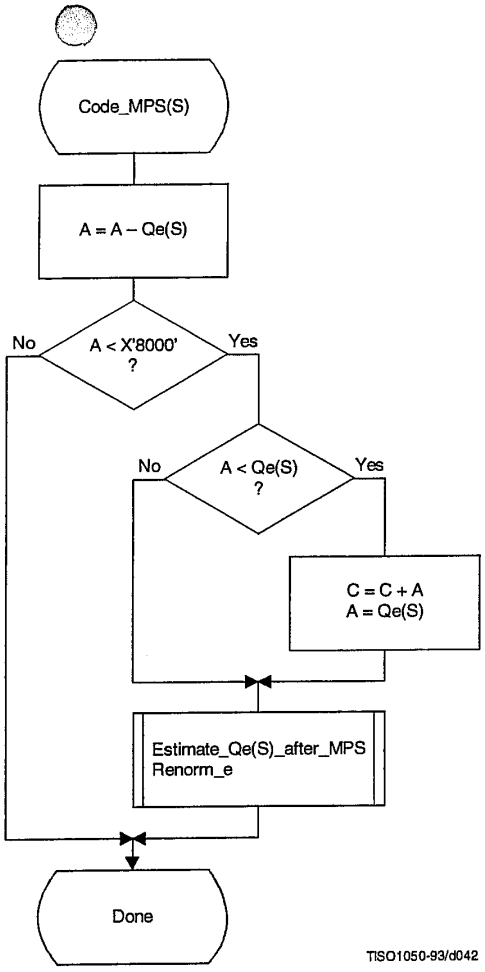


Figure D.4 – Code_MPS(S) procedure with conditional MPS/LPS exchange

D.1.5 Probability estimation in the encoder

D.1.5.1 Probability estimation state machine

The probability estimation state machine consists of a number of sequences of probability estimates. These sequences are interlinked in a manner which provides probability estimates based on approximate symbol counts derived from the arithmetic coder renormalization. Some of these sequences are used during the initial "learning" stages of probability estimation; the rest are used for "steady state" estimation.

Each entry in the probability estimation state machine is assigned an index, and each index has associated with it a Qe value and two Next_Index values. The Next_Index_MPS gives the index to the new probability estimate after an MPS renormalization; the Next_Index_LPS gives the index to the new probability estimate after an LPS renormalization. Note that both the index to the estimation state machine and the sense of the MPS are kept for each context-index S. The sense of the MPS is changed whenever the entry in the Switch_MPS is one.

The probability estimation state machine is given in Table D.3. Initialization of the arithmetic coder is always with an MPS sense of zero and a Qe index of zero in Table D.3.

The Qe values listed in Table D.3 are expressed as hexadecimal integers. To approximately convert the 15-bit integer representation of Qe to a decimal probability, divide the Qe values by (4/3) × (X'8000').

Table D.3 – Qe values and probability estimation state machine

Index	Qe	Next_ Index		Switch	Index	Qe	Next_ Index		Switch
	_Value	_LPS	_MPS	_MPS		_Value	_LPS	_MPS	_MPS
0	X'5A1D'	1	1	1	57	X'01A4'	55	58	0
1	X'2586'	14	2	0	58	X'0160'	56	59	0
2	X'1114'	16	3	0	59	X'0125'	57	60	0
3	X'080B'	18	4	0	60	X'00F6'	58	61	0
4	X'03D8'	20	5	0	61	X'00CB'	59	62	0
5	X'01DA'	23	6	0	62	X'00AB'	61	63	0
6	X'00E5'	25	7	0	63	X'008F'	61	32	0
7	X'006F'	28	8	0	64	X'5B12'	65	65	1
8	X'0036'	30	9	0	65	X'4D04'	80	66	0
9	X'001A'	33	10	0	66	X'412C'	81	67	0
10	X'000D'	35	11	0	67	X'37D8'	82	68	0
11	X'0006'	9	12	0	68	X'2FE8'	83	69	0
12	X'0003'	10	13	0	69	X'293C'	84	70	0
13	X'0001'	12	13	0	70	X'2379'	86	71	0
14	X'5A7F'	15	15	1	71	X'1EDF'	87	72	0
15	X'3F25'	36	16	0	72	X'1AA9'	87	73	0
16	X'2CF2'	38	17	0	73	X'174E'	72	74	0
17	X'207C'	39	18	0	74	X'1424'	72	75	0
18	X'17B9'	40	19	0	75	X'119C'	74	76	0
19	X'1182'	42	20	0	76	X'0F6B'	74	77	0
20	X'0CEF'	43	21	0	77	X'0D51'	75	78	0
21	X'09A1'	45	22	0	78	X'0BB6'	77	79	0
22	X'072F'	46	23	0	79	X'0A40'	77	48	0
23	X'055C'	48	24	0	80	X'5832'	80	81	1
24	X'0406'	49	25	0	81	X'4D1C'	88	82	0
25	X'0303'	51	26	0	82	X'438E'	89	83	0
26	X'0240'	52	27	0	83	X'3BDD'	90	84	0
27	X'01B1'	54	28	0	84	X'34EE'	91	85	0
28	X'0144'	56	29	0	85	X'2EAE'	92	86	0
29	X'00F5'	57	30	0	86	X'299A'	93	87	0
30	X'00B7'	59	31	0	87	X'2516'	86	71	0
31	X'008A'	60	32	0	88	X'5570'	88	89	1
32	X'0068'	62	33	0	89	X'4CA9'	95	90	0
33	X'004E'	63	34	0	90	X'44D9'	96	91	0
34	X'003B'	32	35	0	91	X'3E22'	97	92	0
35	X'002C'	33	9	0	92	X'3824'	99	93	0
36	X'5AE1'	37	37	1	93	X'32B4'	99	94	0
37	X'484C'	64	38	0	94	X'2E17'	93	86	0
38	X'3A0D'	65	39	0	95	X'56A8'	95	96	1
39	X'2EF1'	67	40	0	96	X'4F46'	101	97	0
40	X'261F'	68	41	0	97	X'47E5'	102	98	0
41	X'1F33'	69	42	0	98	X'41CF'	103	99	0
42	X'19A8'	70	43	0	99	X'3C3D'	104	100	0
43	X'1518'	72	44	0	100	X'375E'	99	93	0
44	X'1177'	73	45	0	101	X'5231'	105	102	0
45	X'0E74'	74	46	0	102	X'4C0F'	106	103	0
46	X'0BFB'	75	47	0	103	X'4639'	107	104	0
47	X'09F8'	77	48	0	104	X'415E'	103	99	0
48	X'0861'	78	49	0	105	X'5627'	105	106	1
49	X'0706'	79	50	0	106	X'50E7'	108	107	0
50	X'05CD'	48	51	0	107	X'4B85'	109	103	0
51	X'04DE'	50	52	0	108	X'5597'	110	109	0
52	X'040F'	50	53	0	109	X'504F'	111	107	0
53	X'0363'	51	54	0	110	X'5A10'	110	111	1
54	X'02D4'	52	55	0	111	X'5522'	112	109	0
55	X'025C'	53	56	0	112	X'59EB'	112	111	1
56	X'01F8'	54	57	0					

D.1.5.2 Renormalization driven estimation

The change in state in Table D.3 occurs only when the arithmetic coder interval register is renormalized. This must always be done after coding an LPS, and whenever the probability interval register is less than X'8000' (0.75 in decimal notation) after coding an MPS.

When the LPS renormalization is required, Next_Index_LPS gives the new index for the LPS probability estimate. When the MPS renormalization is required, Next_Index_MPS gives the new index for the LPS probability estimate. If Switch_MPS is 1 for the old index, the MPS symbol sense must be inverted after an LPS.

D.1.5.3 Estimation following renormalization after MPS

The procedure for estimating the probability on the MPS renormalization path is given in Figure D.5. Index(S) is part of the information stored for context-index S. The new value of Index(S) is obtained from Table D.3 from the column labeled Next_Index_MPS, as that is the next index after an MPS renormalization. This next index is stored as the new value of Index(S) in the context storage at context-index S, and the value of Qe at this new Index(S) becomes the new Qe(S). MPS(S) does not change.

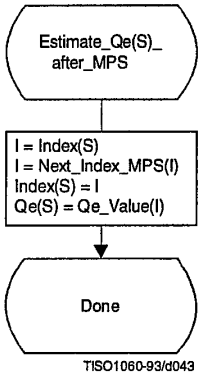


Figure D.5 – Probability estimation on MPS renormalization path

D.1.5.4 Estimation following renormalization after LPS

The procedure for estimating the probability on the LPS renormalization path is shown in Figure D.6. The procedure is similar to that of Figure D.5 except that when Switch_MPS(I) is 1, the sense of MPS(S) must be inverted.

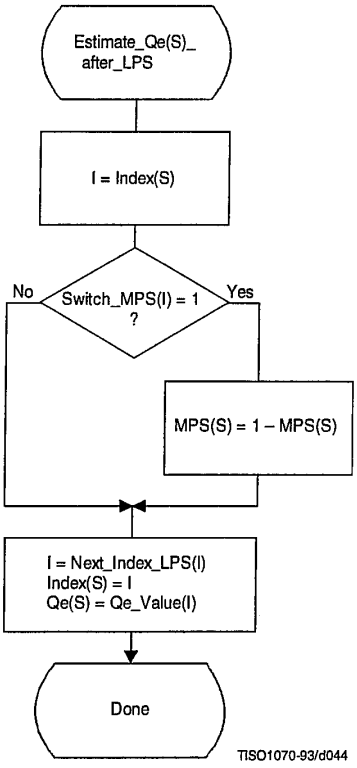


Figure D.6 – Probability estimation on LPS renormalization path

D.1.6 Renormalization in the encoder

The Renorm_e procedure for the encoder renormalization is shown in Figure D.7. Both the probability interval register A and the code register C are shifted, one bit at a time. The number of shifts is counted in the counter CT; when CT is zero, a byte of compressed data is removed from C by the procedure Byte_out and CT is reset to 8. Renormalization continues until A is no longer less than X'8000'.

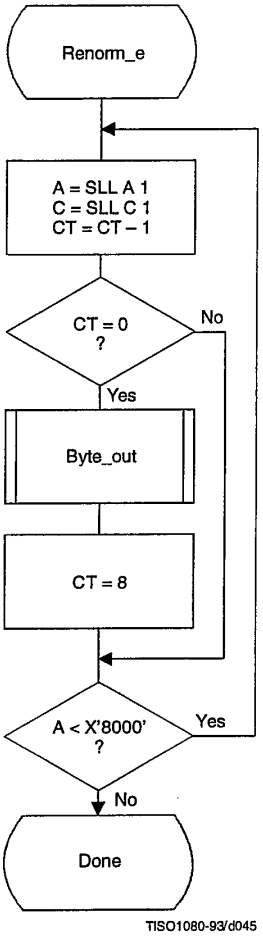


Figure D.7 – Encoder renormalization procedure

The Byte_out procedure used in Renorm_e is shown in Figure D.8. This procedure uses byte-stuffing procedures which prevent accidental generation of markers by the arithmetic encoding procedures. It also includes an example of a procedure for resolving carry-over. For simplicity of exposition, the buffer holding the entropy-coded segment is assumed to be large enough to contain the entire segment.

In Figure D.8 BP is the entropy-coded segment pointer and B is the compressed data byte pointed to by BP. T in Byte_out is a temporary variable which is used to hold the output byte and carry bit. ST is the stack counter which is used to count X'FF' output bytes until any carry-over through the X'FF' sequence has been resolved. The value of ST rarely exceeds 3. However, since the upper limit for the value of ST is bounded only by the total entropy-coded segment size, a precision of 32 bits is recommended for ST.

Since large values of ST represent a latent output of compressed data, the following procedure may be needed in high speed synchronous encoding systems for handling the burst of output data which occurs when the carry is resolved.

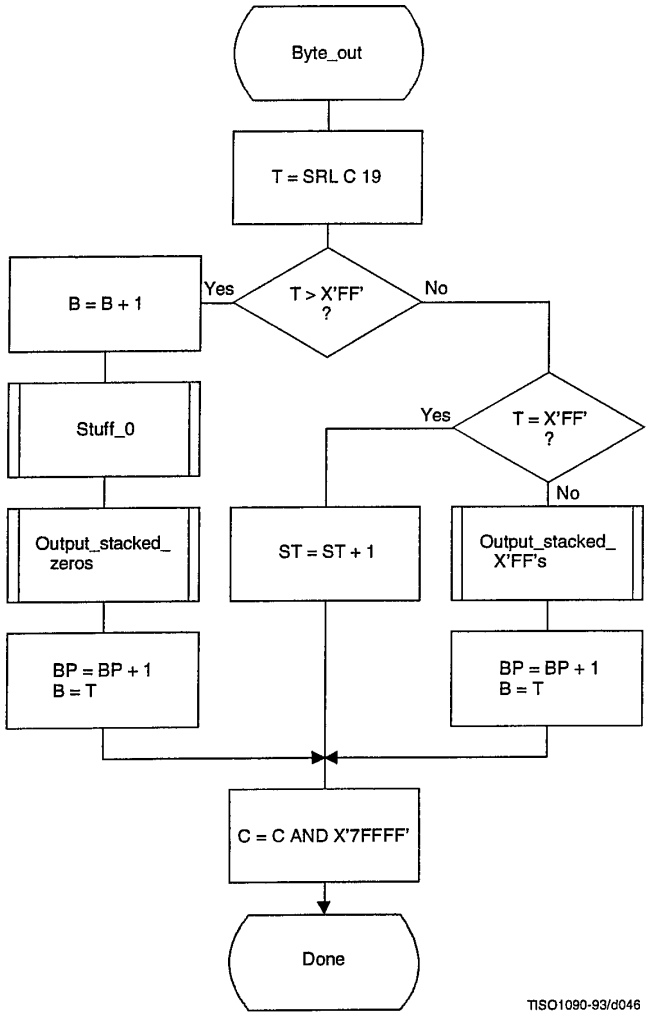


Figure D.8 – Byte_out procedure for encoder

When the stack count reaches an upper bound determined by output channel capacity, the stack is emptied and the stacked X'FF' bytes (and stuffed zero bytes) are added to the compressed data before the carry-over is resolved. If a carry-over then occurs, the carry is added to the final stuffed zero, thereby converting the final X'FF00' sequence to the X'FF01' temporary private marker. The entropy-coded segment must then be post-processed to resolve the carry-over and remove the temporary marker code. For any reasonable bound on ST this post processing is very unlikely.

Referring to Figure D.8, the shift of the code register by 19 bits aligns the output bits with the low order bits of T. The first test then determines if a carry-over has occurred. If so, the carry must be added to the previous output byte before advancing the segment pointer BP. The Stuff_0 procedure stuffs a zero byte whenever the addition of the carry to the data already in the entropy-coded segments creates a X'FF' byte. Any stacked output bytes – converted to zeros by the carry-over – are then placed in the entropy-coded segment. Note that when the output byte is later transferred from T to the entropy-coded segment (to byte B), the carry bit is ignored if it is set.

If a carry has not occurred, the output byte is tested to see if it is X'FF'. If so, the stack count ST is incremented, as the output must be delayed until the carry-over is resolved. If not, the carry-over has been resolved, and any stacked X'FF' bytes must then be placed in the entropy-coded segment. Note that a zero byte is stuffed following each X'FF'.

The procedures used by Byte_out are defined in Figures D.9 through D.11.

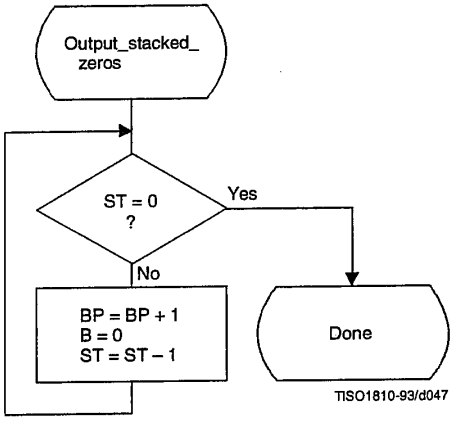


Figure D.9 – Output_stacked_zeros procedure for encoder

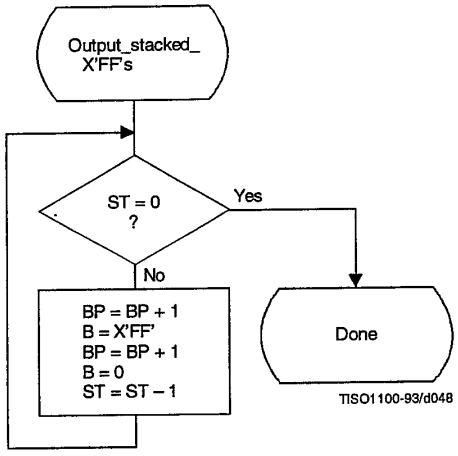


Figure D.10 – Output_stacked_X'FF's procedure for encoder

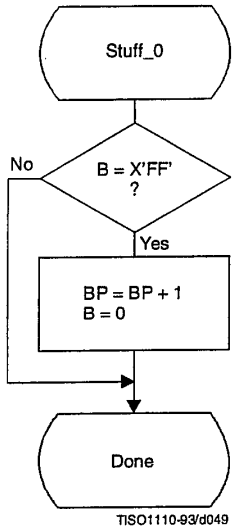


Figure D.11 – Stuff_0 procedure for encoder

D.1.7 Initialization of the encoder

The Initenc procedure is used to start the arithmetic coder. The basic steps are shown in Figure D.12.

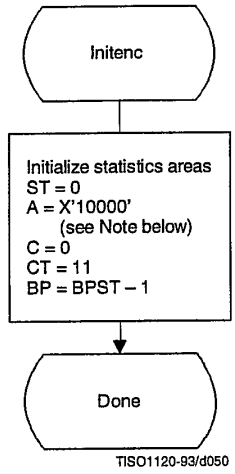


Figure D.12 – Initialization of the encoder

The probability estimation tables are defined by Table D.3. The statistics areas are initialized to an MPS sense of 0 and a Qe index of zero as defined by Table D.3. The stack count (ST) is cleared, the code register (C) is cleared, and the interval register is set to X'10000'. The counter (CT) is set to 11, reflecting the fact that when A is initialized to X'10000' three spacer bits plus eight output bits in C must be filled before the first byte is removed. Note that BP is initialized to point to the byte before the start of the entropy-coded segment (which is at BPST). Note also that the statistics areas are initialized for all values of context-index S to MPS(S) = 0 and Index(S) = 0.

NOTE – Although the probability interval is initialized to X'10000' in both Initenc and Initdec, the precision of the probability interval register can still be limited to 16 bits. When the precision of the interval register is 16 bits, it is initialized to zero.

D.1.8 Termination of encoding

The Flush procedure is used to terminate the arithmetic encoding procedures and prepare the entropy-coded segment for the addition of the X'FF' prefix of the marker which follows the arithmetically coded data. Figure D.13 shows this flush procedure. The first step in the procedure is to set as many low order bits of the code register to zero as possible without pointing outside of the final interval. Then, the output byte is aligned by shifting it left by CT bits; Byte_out then removes it from C. C is then shifted left by 8 bits to align the second output byte and Byte_out is used a second time. The remaining low order bits in C are guaranteed to be zero, and these trailing zero bits shall not be written to the entropy-coded segment.

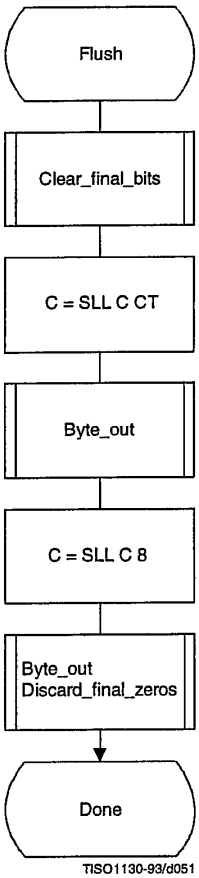


Figure D.13 – Flush procedure

Any trailing zero bytes already written to the entropy-coded segment and not preceded by a X'FF' may, optionally, be discarded. This is done in the Discard_final_zeros procedure. Stuffed zero bytes shall not be discarded.

Entropy coded segments are always followed by a marker. For this reason, the final zero bits needed to complete decoding shall not be included in the entropy coded segment. Instead, when the decoder encounters a marker, zero bits shall be supplied to the decoding procedure until decoding is complete. This convention guarantees that when a DNL marker is used, the decoder will intercept it in time to correctly terminate the decoding procedure.

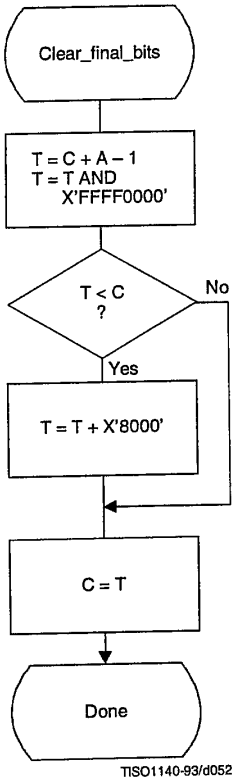


Figure D.14 – Clear_final_bits procedure in Flush

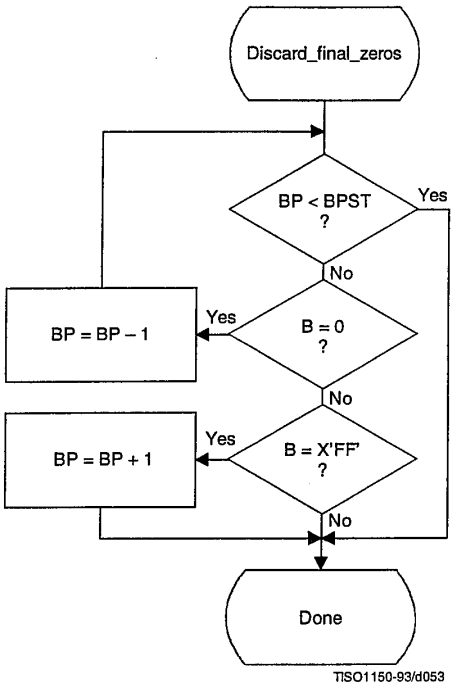


Figure D.15 – Discard_final_zeros procedure in Flush

D.2 Arithmetic decoding procedures

Two arithmetic decoding procedures are used for arithmetic decoding (see Table D.4).

The "Decode(S)" procedure decodes the binary decision for a given context-index S and returns a value of either 0 or 1. It is the inverse of the "Code_0(S)" and "Code_1(S)" procedures described in D.1. "Initdec" initializes the arithmetic coding entropy decoder.

Table D.4 – Procedures for binary arithmetic decoding

Procedure	Purpose
Decode(S)	Decode a binary decision with context-index S
Initdec	Initialize the decoder

D.2.1 Binary arithmetic decoding principles

The probability interval subdivision and sub-interval ordering defined for the arithmetic encoding procedures also apply to the arithmetic decoding procedures.

Since the bit stream always points within the current probability interval, the decoding process is a matter of determining, for each decision, which sub-interval is pointed to by the bit stream. This is done recursively, using the same probability interval sub-division process as in the encoder. Each time a decision is decoded, the decoder subtracts from the bit stream any interval the encoder added to the bit stream. Therefore, the code register in the decoder is a pointer into the current probability interval relative to the base of the interval.

If the size of the sub-interval allocated to the LPS is larger than the sub-interval allocated to the MPS, the encoder invokes the conditional exchange procedure. When the interval sizes are inverted in the decoder, the sense of the symbol decoded must be inverted.

D.2.2 Decoding conventions and approximations

The approximations and integer arithmetic defined for the probability interval subdivision in the encoder must also be used in the decoder. However, where the encoder would have added to the code register, the decoder subtracts from the code register.

D.2.3 Decoder code register conventions

The flow charts given in this section assume the register structures for the decoder as shown in Table D.5:

Table D.5 – Decoder register conventions

	MSB	LSB
Cx register	xxxxxxxx,	xxxxxxxx
C-low	bbbbbbbb,	00000000
A-register	aaaaaaaa,	aaaaaaaa

Cx and C-low can be regarded as one 32-bit C-register, in that renormalization of C shifts a bit of new data from bit 15 of C-low to bit 0 of Cx. However, the decoding comparisons use Cx alone. New data are inserted into the "b" bits of C-low one byte at a time.

NOTE – The comparisons shown in the various procedures use arithmetic comparisons, and therefore assume precisions greater than 16 bits for the variables. Unsigned (logical) comparisons should be used in 16-bit precision implementations.

D.2.4 The decode procedure

The decoder decodes one binary decision at a time. After decoding the decision, the decoder subtracts any amount from the code register that the encoder added. The amount left in the code register is the offset from the base of the current probability interval to the sub-interval allocated to the binary decisions not yet decoded. In the first test in the decode procedure shown in Figure D.16 the code register is compared to the size of the MPS sub-interval. Unless a conditional exchange is needed, this test determines whether the MPS or LPS for context-index S is decoded. Note that the LPS for context-index S is given by 1 – MPS(S).

When a renormalization is needed, the MPS/LPS conditional exchange may also be needed. For the LPS path, the conditional exchange procedure is shown in Figure D.17. Note that the probability estimation in the decoder is identical to the probability estimation in the encoder (Figures D.5 and D.6).

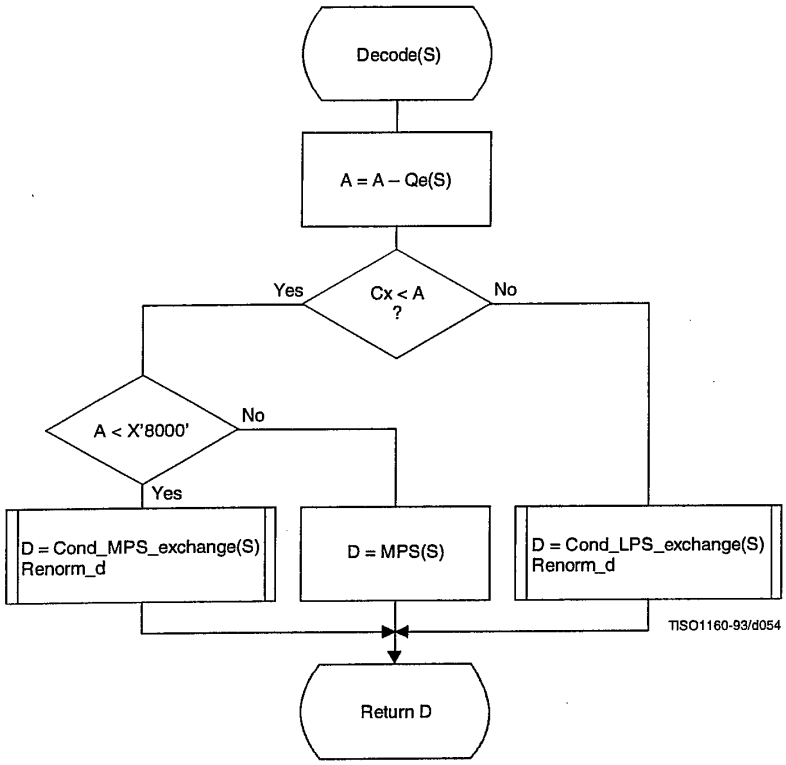


Figure D.16 – Decode(S) procedure

For the MPS path of the decoder the conditional exchange procedure is given in Figure D.18.

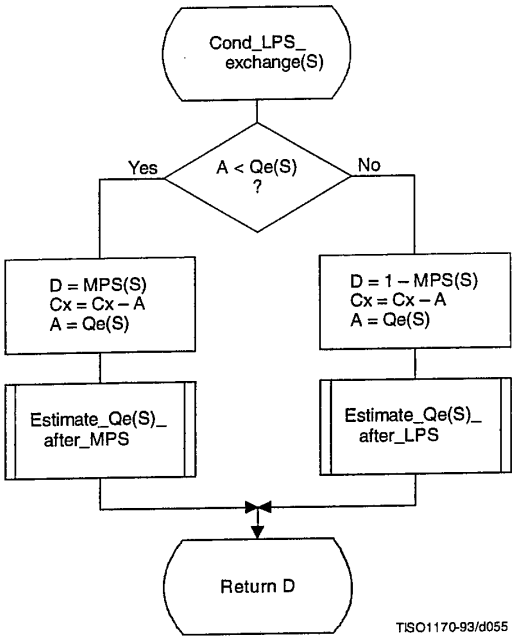


Figure D.17 – Decoder LPS path conditional exchange procedure

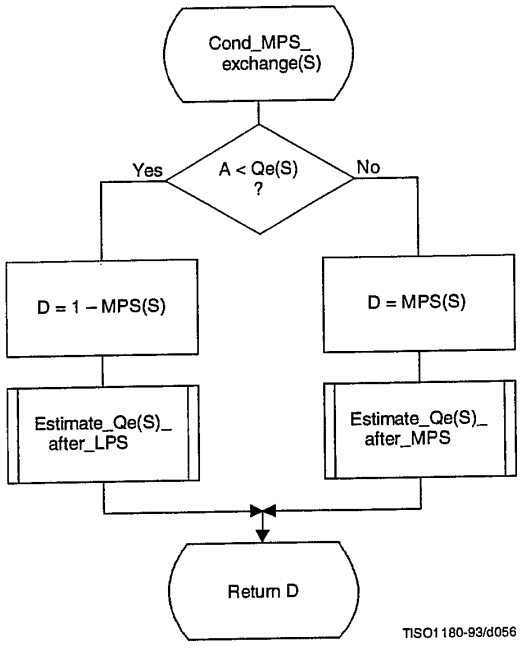


Figure D.18 – Decoder MPS path conditional exchange procedure

D.2.5 Probability estimation in the decoder

The procedures defined for obtaining a new LPS probability estimate in the encoder are also used in the decoder.

D.2.6 Renormalization in the decoder

The Renorm_d procedure for the decoder renormalization is shown in Figure D.19. CT is a counter which keeps track of the number of compressed bits in the C-low section of the C-register. When CT is zero, a new byte is inserted into C-low by the procedure Byte_in and CT is reset to 8.

Both the probability interval register A and the code register C are shifted, one bit at a time, until A is no longer less than X'8000'.

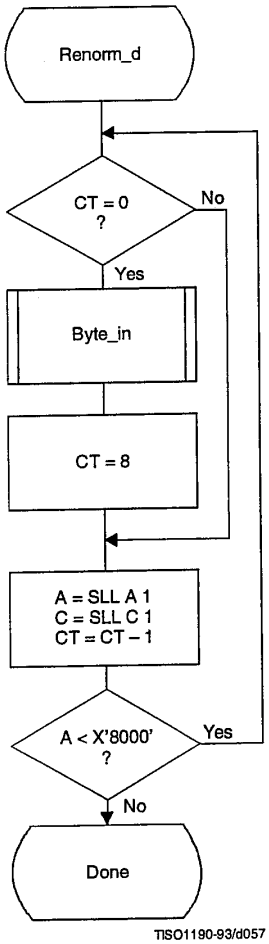


Figure D.19 – Decoder renormalization procedure

The Byte_in procedure used in Renorm_d is shown in Figure D.20. This procedure fetches one byte of data, compensating for the stuffed zero byte which follows any X'FF' byte. It also detects the marker which must follow the entropy-coded segment. The C-register in this procedure is the concatenation of the Cx and C-low registers. For simplicity of exposition, the buffer holding the entropy-coded segment is assumed to be large enough to contain the entire segment.

B is the byte pointed to by the entropy-coded segment pointer BP. BP is first incremented. If the new value of B is not a X'FF', it is inserted into the high order 8 bits of C-low.

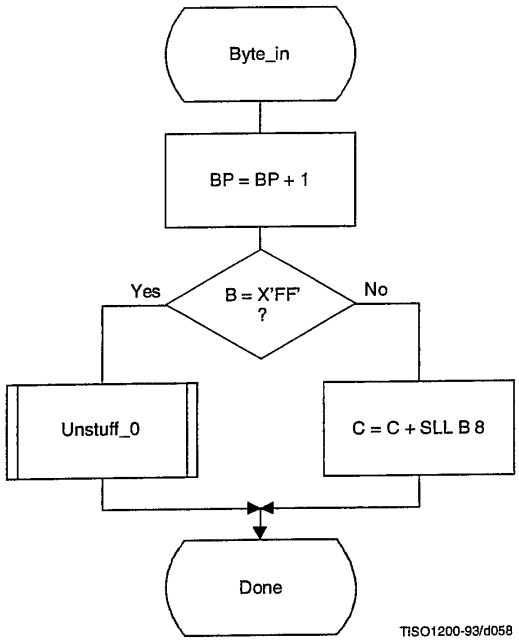


Figure D.20 – Byte_in procedure for decoder

The Unstuff_0 procedure is shown in Figure D.21. If the new value of B is X'FF', BP is incremented to point to the next byte and this next B is tested to see if it is zero. If so, B contains a stuffed byte which must be skipped. The zero B is ignored, and the X'FF' B value which preceded it is inserted in the C-register.

If the value of B after a X'FF' byte is not zero, then a marker has been detected. The marker is interpreted as required and the entropy-coded segment pointer is adjusted ("Adjust BP" in Figure D.21) so that 0-bytes will be fed to the decoder until decoding is complete. One way of accomplishing this is to point BP to the byte preceding the marker which follows the entropy-coded segment.

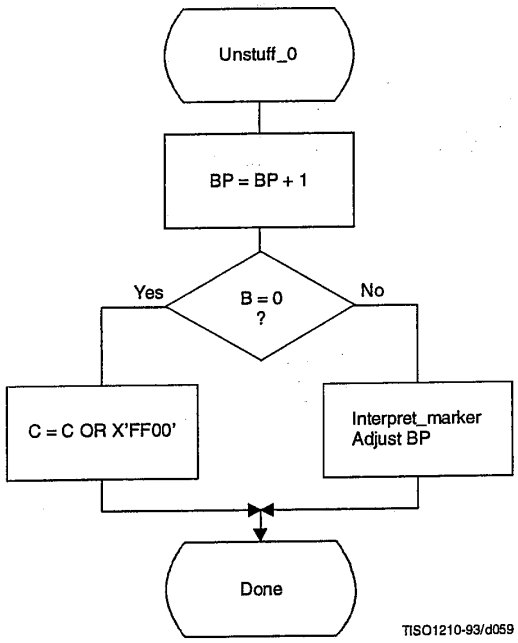


Figure D.21 – Unstuff_0 procedure for decoder

D.2.7 Initialization of the decoder

The Initdec procedure is used to start the arithmetic decoder. The basic steps are shown in Figure D.22.

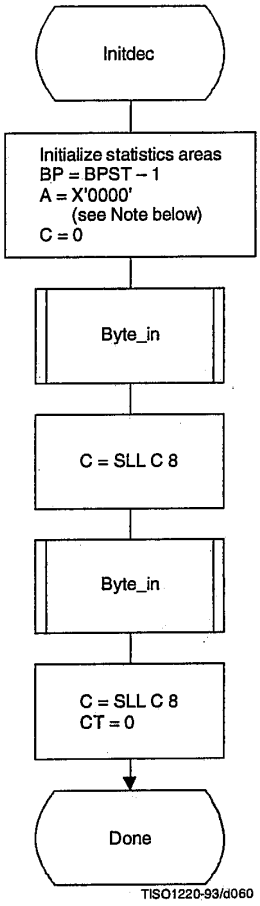


Figure D.22 – Initialization of the decoder

The estimation tables are defined by Table D.3. The statistics areas are initialized to an MPS sense of 0 and a Qe index of zero as defined by Table D.3. BP, the pointer to the entropy-coded segment, is then initialized to point to the byte before the start of the entropy-coded segment at BPST, and the interval register is set to the same starting value as in the encoder. The first byte of compressed data is fetched and shifted into Cx. The second byte is then fetched and shifted into Cx. The count is set to zero, so that a new byte of data will be fetched by Renorm_d.

NOTE – Although the probability interval is initialized to X'10000' in both Initenc and Initdec, the precision of the probability interval register can still be limited to 16 bits. When the precision of the interval register is 16 bits, it is initialized to zero.

D.3 Bit ordering within bytes

The arithmetically encoded entropy-coded segment is an integer of variable length. Therefore, the ordering of bytes and the bit ordering within bytes is the same as for parameters (see B.1.1.1).

Annex E

Encoder and decoder control procedures

(This annex forms an integral part of this Recommendation I International Standard)

This annex describes the encoder and decoder control procedures for the sequential, progressive, and lossless modes of operation.

The encoding and decoding control procedures for the hierarchical processes are specified in Annex J.

NOTES

1 There is **no requirement** in this Specification that any encoder or decoder shall implement the procedures in precisely the manner specified by the flow charts in this annex. It is necessary only that an encoder or decoder implement the **function** specified in this annex. The sole criterion for an encoder or decoder to be considered in compliance with this Specification is that it satisfy the requirements given in clause 6 (for encoders) or clause 7 (for decoders), as determined by the compliance tests specified in Part 2.

2 Implementation-specific setup steps are not indicated in this annex and may be necessary.

E.1 Encoder control procedures

E.1.1 Control procedure for encoding an image

The encoder control procedure for encoding an image is shown in Figure E.1.

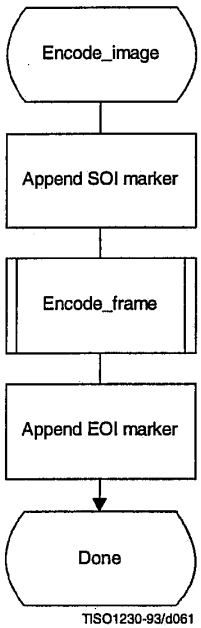


Figure E.1 – Control procedure for encoding an image

E.1.2 Control procedure for encoding a frame

In all cases where markers are appended to the compressed data, optional X'FF' fill bytes may precede the marker.

The control procedure for encoding a frame is oriented around the scans in the frame. The frame header is first appended, and then the scans are coded. Table specifications and other marker segments may precede the SOF_n marker, as indicated by [tables/miscellaneous] in Figure E.2.

Figure E.2 shows the encoding process frame control procedure.

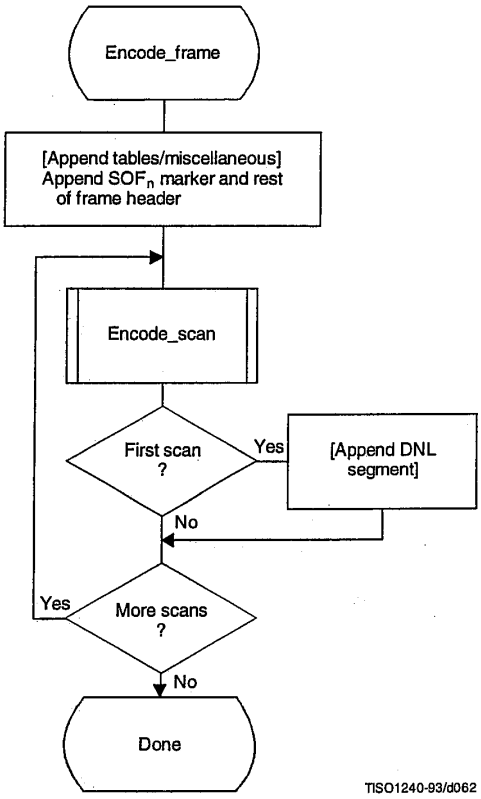


Figure E.2 – Control procedure for encoding a frame

E.1.3 Control procedure for encoding a scan

A scan consists of a single pass through the data of each component in the scan. Table specifications and other marker segments may precede the SOS marker. If more than one component is coded in the scan, the data are interleaved. If restart is enabled, the data are segmented into restart intervals. If restart is enabled, a RST_m marker is placed in the coded data between restart intervals. If restart is disabled, the control procedure is the same, except that the entire scan contains a single restart interval. The compressed image data generated by a scan is always followed by a marker, either the EOI marker or the marker of the next marker segment.

Figure E.3 shows the encoding process scan control procedure. The loop is terminated when the encoding process has coded the number of restart intervals which make up the scan. "m" is the restart interval modulo counter needed for the RST_m marker. The modulo arithmetic for this counter is shown after the "Append RST_m marker" procedure.

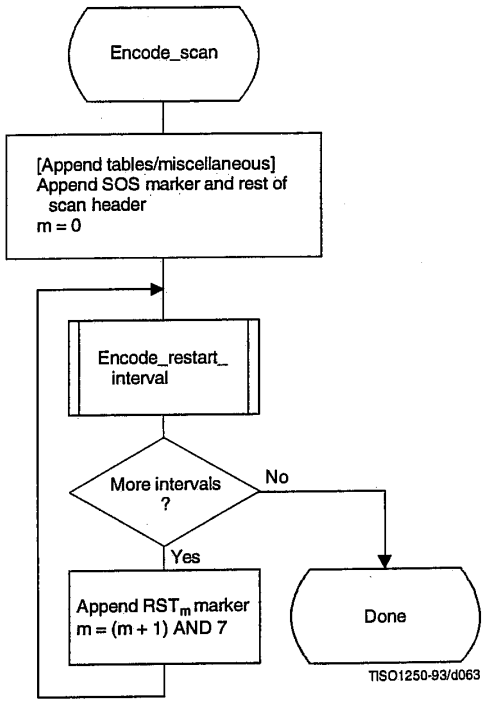


Figure E.3 – Control procedure for encoding a scan

E.1.4 Control procedure for encoding a restart interval

Figure E.4 shows the encoding process control procedure for a restart interval. The loop is terminated either when the encoding process has coded the number of minimum coded units (MCU) in the restart interval or when it has completed the image scan.

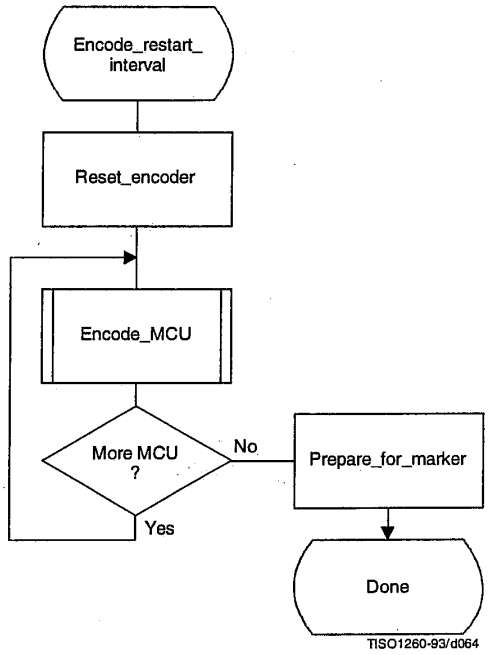


Figure E.4 – Control procedure for encoding a restart interval

The "Reset_encoder" procedure consists at least of the following:

a) if arithmetic coding is used, initialize the arithmetic encoder using the "Initenc" procedure described in D.1.7;

b) for DCT-based processes, set the DC prediction (PRED) to zero for all components in the scan (see F.1.1.5.1);

c) for lossless processes, reset the prediction to a default value for all components in the scan (see H.1.1);

d) do all other implementation-dependent setups that may be necessary.

The procedure "Prepare_for_marker" terminates the entropy-coded segment by:

a) padding a Huffman entropy-coded segment with 1-bits to complete the final byte (and if needed stuffing a zero byte) (see F.1.2.3); or

b) invoking the procedure "Flush" (see D.1.8) to terminate an arithmetic entropy-coded segment.

NOTE – The number of minimum coded units (MCU) in the final restart interval must be adjusted to match the number of MCU in the scan. The number of MCU is calculated from the frame and scan parameters. (See Annex B.)

E.1.5 Control procedure for encoding a minimum coded unit (MCU)

The minimum coded unit is defined in A.2. Within a given MCU the data units are coded in the order in which they occur in the MCU. The control procedure for encoding a MCU is shown in Figure E.5.

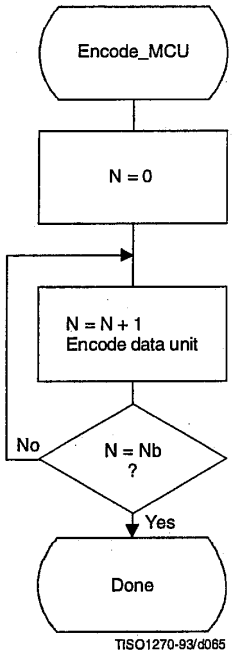


Figure E.5 – Control procedure for encoding a minimum coded unit (MCU)

In Figure E.5, Nb refers to the number of data units in the MCU. The order in which data units occur in the MCU is defined in A.2. The data unit is an 8×8 block for DCT-based processes, and a single sample for lossless processes.

The procedures for encoding a data unit are specified in Annexes F, G, and H.

E.2 Decoder control procedures

E.2.1 Control procedure for decoding compressed image data

Figure E.6 shows the decoding process control for compressed image data.

Decoding control centers around identification of various markers. The first marker must be the SOI (Start Of Image) marker. The "Decoder_setup" procedure resets the restart interval (Ri = 0) and, if the decoder has arithmetic decoding capabilities, sets the conditioning tables for the arithmetic coding to their default values. (See F.1.4.4.1.4 and F.1.4.4.2.1.) The next marker is normally a SOF_n (Start Of Frame) marker; if this is not found, one of the marker segments listed in Table E.1 has been received.

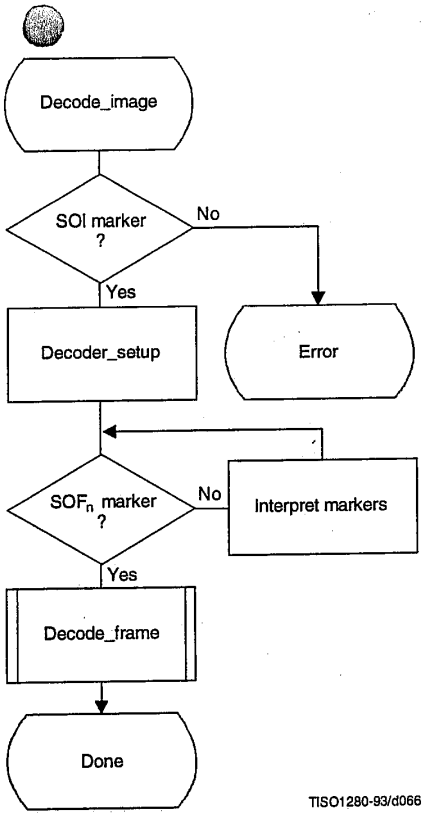


Figure E.6 – Control procedure for decoding compressed image data

Table E.1 – Markers recognized by "Interpret markers"

Marker	Purpose
DHT	Define Huffman Tables
DAC	Define Arithmetic Conditioning
DQT	Define Quantization Tables
DRI	Define Restart Interval
APP$_n$	Application defined marker
COM	Comment

Note that optional X'FF' fill bytes which may precede any marker shall be discarded before determining which marker is present.

The additional logic to interpret these various markers is contained in the box labeled "Interpret markers". DHT markers shall be interpreted by processes using Huffman coding. DAC markers shall be interpreted by processes using arithmetic coding. DQT markers shall be interpreted by DCT-based decoders. DRI markers shall be interpreted by all decoders. APPn and COM markers shall be interpreted only to the extent that they do not interfere with the decoding.

By definition, the procedures in "Interpret markers" leave the system at the next marker. Note that if the expected SOI marker is missing at the start of the compressed image data, an error condition has occurred. The techniques for detecting and managing error conditions can be as elaborate or as simple as desired.

E.2.2 Control procedure for decoding a frame

Figure E.7 shows the control procedure for the decoding of a frame.

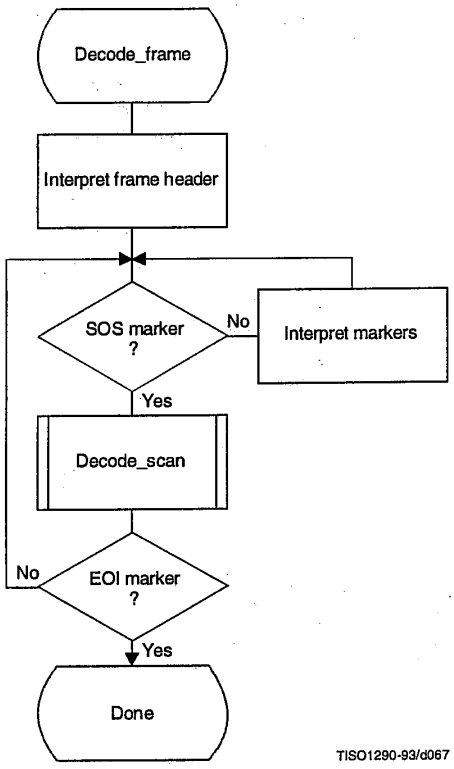


Figure E.7 – Control procedure for decoding a frame

The loop is terminated if the EOI marker is found at the end of the scan.

The markers recognized by "Interpret markers" are listed in Table E.1. Subclause E.2.1 describes the extent to which the various markers shall be interpreted.

E.2.3 Control procedure for decoding a scan

Figure E.8 shows the decoding of a scan.

The loop is terminated when the expected number of restart intervals has been decoded.

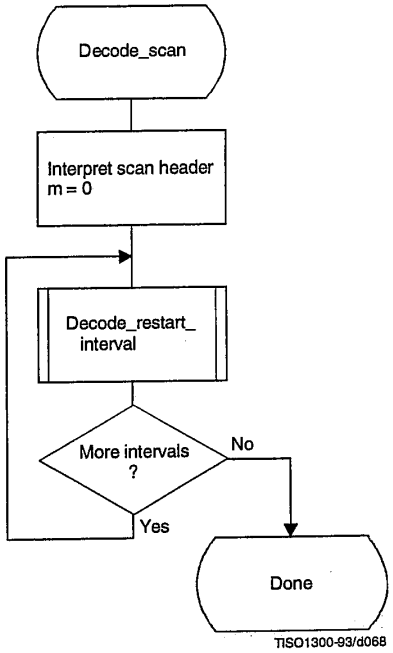


Figure E.8 – Control procedure for decoding a scan

E.2.4 Control procedure for decoding a restart interval

The procedure for decoding a restart interval is shown in Figure E.9. The "Reset_decoder" procedure consists at least of the following:

a) if arithmetic coding is used, initialize the arithmetic decoder using the "Initdec" procedure described in D.2.7;

b) for DCT-based processes, set the DC prediction (PRED) to zero for all components in the scan (see F.2.1.3.1);

c) for lossless process, reset the prediction to a default value for all components in the scan (see H.2.1);

d) do all other implementation-dependent setups that may be necessary.

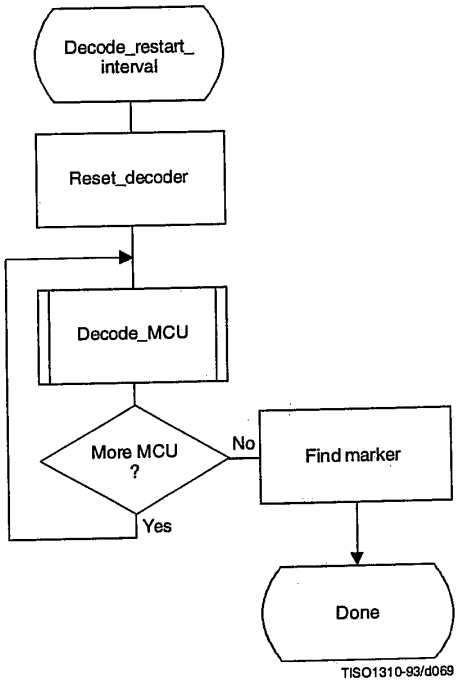


Figure E.9 – Control procedure for decoding a restart interval

At the end of the restart interval, the next marker is located. If a problem is detected in locating this marker, error handling procedures may be invoked. While such procedures are optional, the decoder shall be able to correctly recognize restart markers in the compressed data and reset the decoder when they are encountered. The decoder shall also be able to recognize the DNL marker, set the number of lines defined in the DNL segment, and end the "Decode_restart_interval" procedure.

NOTE – The final restart interval may be smaller than the size specified by the DRI marker segment, as it includes only the number of MCUs remaining in the scan.

E.2.5 Control procedure for decoding a minimum coded unit (MCU)

The procedure for decoding a minimum coded unit (MCU) is shown in Figure E.10.

In Figure E.10 Nb is the number of data units in a MCU.

The procedures for decoding a data unit are specified in Annexes F, G, and H.

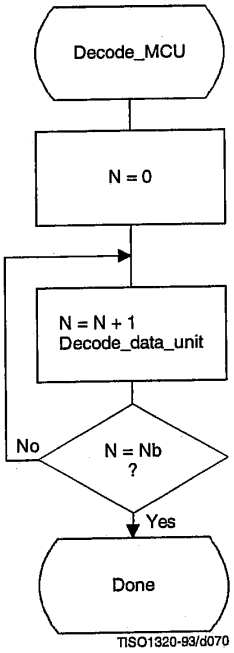


Figure E.10 – Control procedure for decoding a minimum coded unit (MCU)

Annex F

Sequential DCT-based mode of operation

(This annex forms an integral part of this Recommendation | International Standard)

This annex provides a **functional specification** of the following coding processes for the sequential DCT-based mode of operation:

1) baseline sequential;
2) extended sequential, Huffman coding, 8-bit sample precision;
3) extended sequential, arithmetic coding, 8-bit sample precision;
4) extended sequential, Huffman coding, 12-bit sample precision;
5) extended sequential, arithmetic coding, 12-bit sample precision.

For each of these, the encoding process is specified in F.1, and the decoding process is specified in F.2. The functional specification is presented by means of specific flow charts for the various procedures which comprise these coding processes.

NOTE – There is **no requirement** in this Specification that any encoder or decoder which embodies one of the above-named processes shall implement the procedures in precisely the manner specified by the flow charts in this annex. It is necessary only that an encoder or decoder implement the **function** specified in this annex. The sole criterion for an encoder or decoder to be considered in compliance with this Specification is that it satisfy the requirements given in clause 6 (for encoders) or clause 7 (for decoders), as determined by the compliance tests specified in Part 2.

F.1 Sequential DCT-based encoding processes

F.1.1 Sequential DCT-based control procedures and coding models

F.1.1.1 Control procedures for sequential DCT-based encoders

The control procedures for encoding an image and its constituent parts – the frame, scan, restart interval and MCU – are given in Figures E.1 to E.5. The procedure for encoding a MCU (see Figure E.5) repetitively calls the procedure for encoding a data unit. For DCT-based encoders the data unit is an 8×8 block of samples.

F.1.1.2 Procedure for encoding an 8×8 block data unit

For the sequential DCT-based processes encoding an 8×8 block data unit consists of the following procedures:

a) level shift, calculate forward 8×8 DCT and quantize the resulting coefficients using table destination specified in frame header;

b) encode DC coefficient for 8×8 block using DC table destination specified in scan header;

c) encode AC coefficients for 8×8 block using AC table destination specified in scan header.

F.1.1.3 Level shift and forward DCT (FDCT)

The mathematical definition of the FDCT is given in A.3.3.

Prior to computing the FDCT the input data are level shifted to a signed two's complement representation as described in A.3.1. For 8-bit input precision the level shift is achieved by subtracting 128. For 12-bit input precision the level shift is achieved by subtracting 2048.

F.1.1.4 Quantization of the FDCT

The uniform quantization procedure described in Annex A is used to quantize the DCT coefficients. One of four quantization tables may be used by the encoder. No default quantization tables are specified in this Specification. However, some typical quantization tables are given in Annex K.

The quantized DCT coefficient values are signed, two's complement integers with 11-bit precision for 8-bit input precision and 15-bit precision for 12-bit input precision.

F.1.1.5 Encoding models for the sequential DCT procedures

The two dimensional array of quantized DCT coefficients is rearranged in a zig-zag sequence order defined in A.3.6. The zig-zag order coefficients are denoted ZZ (0) through ZZ(63) with:

$$ZZ(0) = Sq_{00}, ZZ(1) = Sq_{01}, ZZ(2) = Sq_{10}, \bullet, \bullet, \bullet, ZZ(63) = Sq_{77}$$

Sq_{vu} are defined in Figure A.6.

Two coding procedures are used, one for the DC coefficient ZZ(0) and the other for the AC coefficients ZZ(1)..ZZ(63). The coefficients are encoded in the order in which they occur in zig-zag sequence order, starting with the DC coefficient. The coefficients are represented as two's complement integers.

F.1.1.5.1 Encoding model for DC coefficients

The DC coefficients are coded differentially, using a one-dimensional predictor, PRED, which is the quantized DC value from the most recently coded 8×8 block from the same component. The difference, DIFF, is obtained from

$$DIFF = ZZ(0) - PRED$$

At the beginning of the scan and at the beginning of each restart interval, the prediction for the DC coefficient prediction is initialized to 0. (Recall that the input data have been level shifted to two's complement representation.)

F.1.1.5.2 Encoding model for AC coefficients

Since many coefficients are zero, runs of zeros are identified and coded efficiently. In addition, if the remaining coefficients in the zig-zag sequence order are all zero, this is coded explicitly as an end-of-block (EOB).

F.1.2 Baseline Huffman encoding procedures

The baseline encoding procedure is for 8-bit sample precision. The encoder may employ up to two DC and two AC Huffman tables within one scan.

F.1.2.1 Huffman encoding of DC coefficients

F.1.2.1.1 Structure of DC code table

The DC code table consists of a set of Huffman codes (maximum length 16 bits) and appended additional bits (in most cases) which can code any possible value of DIFF, the difference between the current DC coefficient and the prediction. The Huffman codes for the difference categories are generated in such a way that no code consists entirely of 1-bits (X'FF' prefix marker code avoided).

The two's complement difference magnitudes are grouped into 12 categories, SSSS, and a Huffman code is created for each of the 12 difference magnitude categories (see Table F.1).

For each category, except SSSS = 0, an additional bits field is appended to the code word to uniquely identify which difference in that category actually occurred. The number of extra bits is given by SSSS; the extra bits are appended to the LSB of the preceding Huffman code, most significant bit first. When DIFF is positive, the SSSS low order bits of DIFF are appended. When DIFF is negative, the SSSS low order bits of (DIFF – 1) are appended. Note that the most significant bit of the appended bit sequence is 0 for negative differences and 1 for positive differences.

F.1.2.1.2 Defining Huffman tables for the DC coefficients

The syntax for specifying the Huffman tables is given in Annex B. The procedure for creating a code table from this information is described in Annex C. No more than two Huffman tables may be defined for coding of DC coefficients. Two examples of Huffman tables for coding of DC coefficients are provided in Annex K.

Table F.1 – Difference magnitude categories for DC coding

SSSS	DIFF values
0	0
1	–1,1
2	–3,–2,2,3
3	–7..–4,4..7
4	–15..–8,8..15
5	–31..–16,16..31
6	–63..–32,32..63
7	–127..–64,64..127
8	–255..–128,128..255
9	–511..–256,256..511
10	–1 023..–512,512..1 023
11	–2 047..–1 024,1 024..2 047

F.1.2.1.3 Huffman encoding procedures for DC coefficients

The encoding procedure is defined in terms of a set of extended tables, XHUFCO and XHUFSI, which contain the complete set of Huffman codes and sizes for all possible difference values. For full 12-bit precision the tables are relatively large. For the baseline system, however, the precision of the differences may be small enough to make this description practical.

XHUFCO and XHUFSI are generated from the encoder tables EHUFCO and EHUFSI (see Annex C) by appending to the Huffman codes for each difference category the additional bits that completely define the difference. By definition, XHUFCO and XHUFSI have entries for each possible difference value. XHUFCO contains the concatenated bit pattern of the Huffman code and the additional bits field; XHUFSI contains the total length in bits of this concatenated bit pattern. Both are indexed by DIFF, the difference between the DC coefficient and the prediction.

The Huffman encoding procedure for the DC difference, DIFF, is:

SIZE = XHUFSI(DIFF)

CODE = XHUFCO(DIFF)

code SIZE bits of CODE

where DC is the quantized DC coefficient value and PRED is the predicted quantized DC value. The Huffman code (CODE) (including any additional bits) is obtained from XHUFCO and SIZE (length of the code including additional bits) is obtained from XHUFSI, using DIFF as the index to the two tables.

F.1.2.2 Huffman encoding of AC coefficients

F.1.2.2.1 Structure of AC code table

Each non-zero AC coefficient in ZZ is described by a composite 8-bit value, RS, of the form

RS = binary 'RRRRSSSS'

The 4 least significant bits, 'SSSS', define a category for the amplitude of the next non-zero coefficient in ZZ, and the 4 most significant bits, 'RRRR', give the position of the coefficient in ZZ relative to the previous non-zero coefficient (i.e. the run-length of zero coefficients between non-zero coefficients). Since the run length of zero coefficients may exceed 15, the value 'RRRRSSSS' = X'F0' is defined to represent a run length of 15 zero coefficients followed by a coefficient of zero amplitude. (This can be interpreted as a run length of 16 zero coefficients.) In addition, a special value 'RRRRSSSS' = '00000000' is used to code the end-of-block (EOB), when all remaining coefficients in the block are zero.

The general structure of the code table is illustrated in Figure F.1. The entries marked "N/A" are undefined for the baseline procedure.

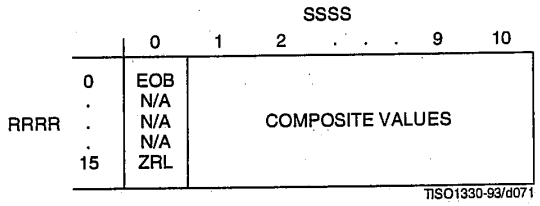


Figure F.1 – Two-dimensional value array for Huffman coding

The magnitude ranges assigned to each value of SSSS are defined in Table F.2.

Table F.2 – Categories assigned to coefficient values

SSSS	AC coefficients
1	–1,1
2	–3,–2,2,3
3	–7..–4,4..7
4	–15..–8,8..15
5	–31..–16,16..31
6	–63..–32,32..63
7	–127..–64,64..127
8	–255..–128,128..255
9	–511..–256,256..511
10	–1 023..–512,512..1 023

The composite value, RRRRSSSS, is Huffman coded and each Huffman code is followed by additional bits which specify the sign and exact amplitude of the coefficient.

The AC code table consists of one Huffman code (maximum length 16 bits, not including additional bits) for each possible composite value. The Huffman codes for the 8-bit composite values are generated in such a way that no code consists entirely of 1-bits.

The format for the additional bits is the same as in the coding of the DC coefficients. The value of SSSS gives the number of additional bits required to specify the sign and precise amplitude of the coefficient. The additional bits are either the low-order SSSS bits of ZZ(K) when ZZ(K) is positive or the low-order SSSS bits of ZZ(K) – 1 when ZZ(K) is negative. ZZ(K) is the *K*th coefficient in the zig-zag sequence of coefficients being coded.

F.1.2.2.2 Defining Huffman tables for the AC coefficients

The syntax for specifying the Huffman tables is given in Annex B. The procedure for creating a code table from this information is described in Annex C.

In the baseline system no more than two Huffman tables may be defined for coding of AC coefficients. Two examples of Huffman tables for coding of AC coefficients are provided in Annex K.

F.1.2.2.3 Huffman encoding procedures for AC coefficients

As defined in Annex C, the Huffman code table is assumed to be available as a pair of tables, EHUFCO (containing the code bits) and EHUFSI (containing the length of each code in bits), both indexed by the composite value defined above.

The procedure for encoding the AC coefficients in a block is shown in Figures F.2 and F.3. In Figure F.2, K is the index to the zig-zag scan position and R is the run length of zero coefficients.

The procedure "Append EHUFSI(X'F0') bits of EHUFCO(X'F0')" codes a run of 16 zero coefficients (ZRL code of Figure F.1). The procedure "Code EHUFSI(0) bits of EHUFCO(0)" codes the end-of-block (EOB code). If the last coefficient (K = 63) is not zero, the EOB code is bypassed.

CSIZE is a procedure which maps an AC coefficient to the SSSS value as defined in Table F.2.

F.1.2.3 Byte stuffing

In order to provide code space for marker codes which can be located in the compressed image data without decoding, byte stuffing is used.

Whenever, in the course of normal encoding, the byte value X'FF' is created in the code string, a X'00' byte is stuffed into the code string.

If a X'00' byte is detected after a X'FF' byte, the decoder must discard it. If the byte is not zero, a marker has been detected, and shall be interpreted to the extent needed to complete the decoding of the scan.

Byte alignment of markers is achieved by padding incomplete bytes with 1-bits. If padding with 1-bits creates a X'FF' value, a zero byte is stuffed before adding the marker.

F.1.3 Extended sequential DCT-based Huffman encoding process for 8-bit sample precision

This process is identical to the Baseline encoding process described in F.1.2, with the exception that the number of sets of Huffman table destinations which may be used within the same scan is increased to four. Four DC and four AC Huffman table destinations is the maximum allowed by this Specification.

F.1.4 Extended sequential DCT-based arithmetic encoding process for 8-bit sample precision

This subclause describes the use of arithmetic coding procedures in the sequential DCT-based encoding process.

NOTE – The arithmetic coding procedures in this Specification are defined for the maximum precision to encourage interchangeability.

The arithmetic coding extensions have the same DCT model as the Baseline DCT encoder. Therefore, Annex F.1.1 also applies to arithmetic coding. As with the Huffman coding technique, the binary arithmetic coding technique is lossless. It is possible to transcode between the two systems without either FDCT or IDCT computations, and without modification of the reconstructed image.

The basic principles of adaptive binary arithmetic coding are described in Annex D. Up to four DC and four AC conditioning table destinations and associated statistics areas may be used within one scan.

The arithmetic encoding procedures for encoding binary decisions, initializing the statistics area, initializing the encoder, terminating the code string, and adding restart markers are listed in Table D.1 of Annex D.

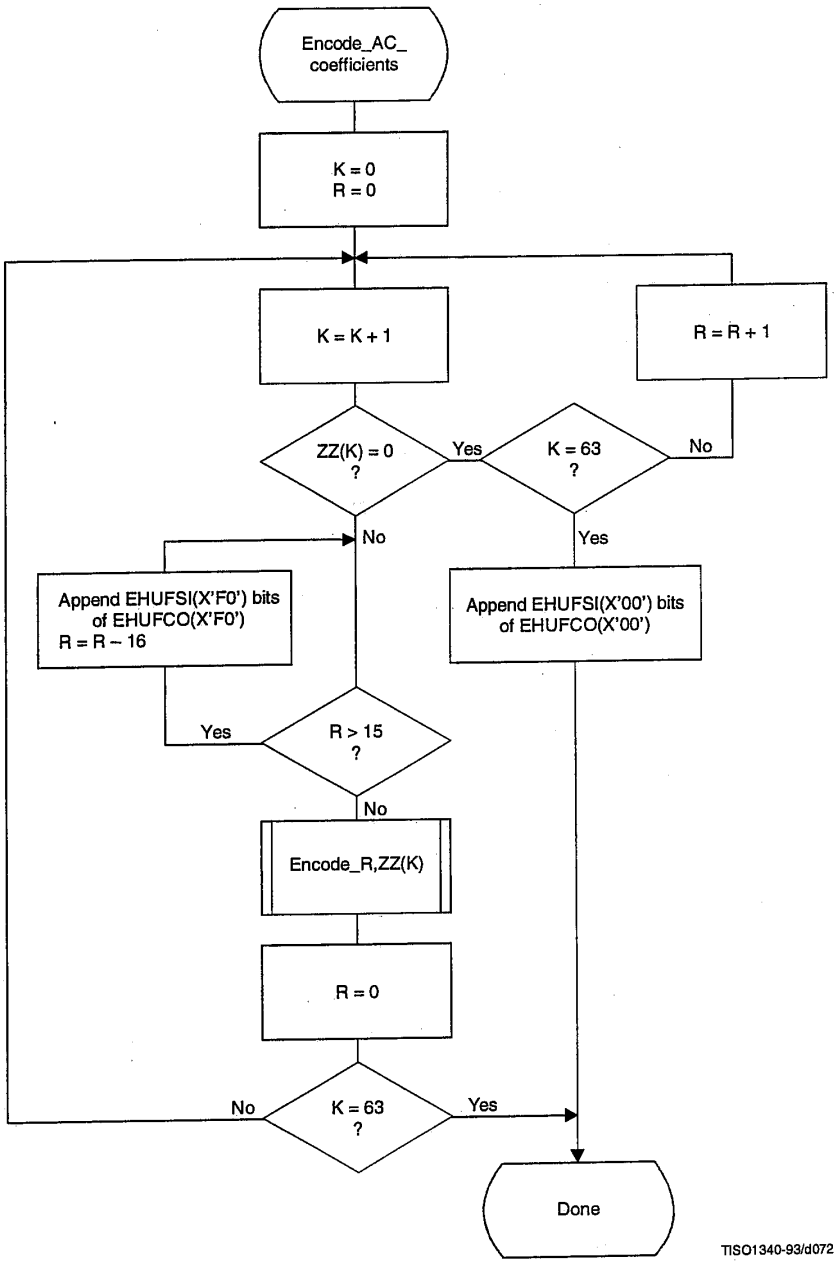


Figure F.2 – Procedure for sequential encoding of AC coefficients with Huffman coding

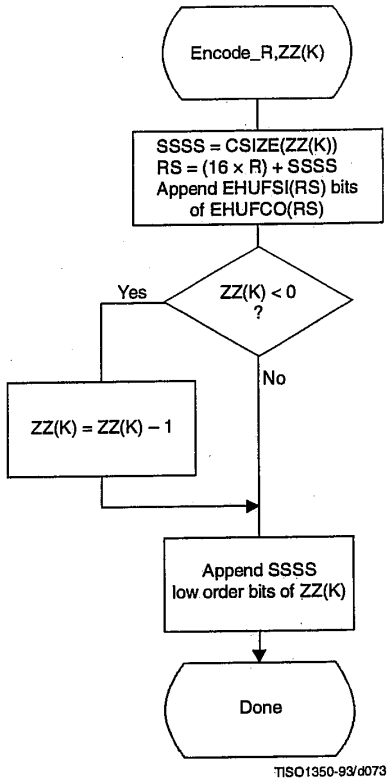


Figure F.3 – Sequential encoding of a non-zero AC coefficient

Some of the procedures in Table D.1 are used in the higher level control structure for scans and restart intervals described in Annex E. At the beginning of scans and restart intervals, the probability estimates used in the arithmetic coder are reset to the standard initial value as part of the Initenc procedure which restarts the arithmetic coder. At the end of scans and restart intervals, the Flush procedure is invoked to empty the code register before the next marker is appended.

F.1.4.1 Arithmetic encoding of DC coefficients

The basic structure of the decision sequence for encoding a DC difference value, DIFF, is shown in Figure F.4.

The context-index S0 and other context-indices used in the DC coding procedures are defined in Table F.4 (see F.1.4.4.1.3). A 0-decision is coded if the difference value is zero and a 1-decision is coded if the difference is not zero. If the difference is not zero, the sign and magnitude are coded using the procedure Encode_V(S0), which is described in F.1.4.3.1.

F.1.4.2 Arithmetic encoding of AC coefficients

The AC coefficients are coded in the order in which they occur in the zig-zag sequence ZZ(1,...,63). An end-of-block (EOB) binary decision is coded before coding the first AC coefficient in ZZ, and after each non-zero coefficient. If the EOB occurs, all remaining coefficients in ZZ are zero. Figure F.5 illustrates the decision sequence. The equivalent procedure for the Huffman coder is found in Figure F.2.

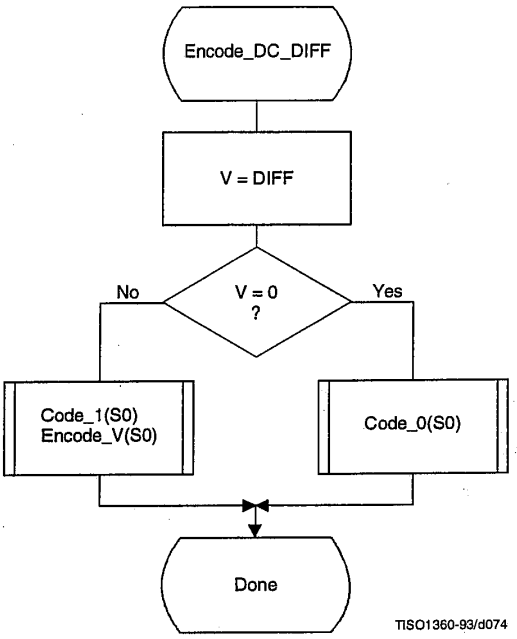


Figure F.4 – Coding model for arithmetic coding of DC difference

The context-indices SE and S0 used in the AC coding procedures are defined in Table F.5 (see F.1.4.4.2). In Figure F.5, K is the index to the zig-zag sequence position. For the sequential scan, Kmin is 1 and Se is 63. The V = 0 decision is part of a loop which codes runs of zero coefficients. Whenever the coefficient is non-zero, "Encode_V(S0)" codes the sign and magnitude of the coefficient. Each time a non-zero coefficient is coded, it is followed by an EOB decision. If the EOB occurs, a 1-decision is coded to indicate that the coding of the block is complete. If the coefficient for K = Se is not zero, the EOB decision is skipped.

F.1.4.3 Encoding the binary decision sequence for non-zero DC differences and AC coefficients

Both the DC difference and the AC coefficients are represented as signed two's complement integer values. The decomposition of these signed integer values into a binary decision tree is done in the same way for both the DC and AC coding models.

Although the binary decision trees for this section of the DC and AC coding models are the same, the statistical models for assigning statistics bins to the binary decisions in the tree are quite different.

F.1.4.3.1 Structure of the encoding decision sequence

The encoding sequence can be separated into three procedures, a procedure which encodes the sign, a second procedure which identifies the magnitude category, and a third procedure which identifies precisely which magnitude occurred within the category identified in the second procedure.

At the point where the binary decision sequence in Encode_V(S0) starts, the coefficient or difference has already been determined to be non-zero. That determination was made in the procedures in Figures F.4 and F.5.

Denoting either DC differences (DIFF) or AC coefficients as V, the non-zero signed integer value of V is encoded by the sequence shown in Figure F.6. This sequence first codes the sign of V. It then (after converting V to a magnitude and decrementing it by 1 to give Sz) codes the magnitude category of Sz (code_log2_Sz), and then codes the low order magnitude bits (code_Sz_bits) to identify the exact magnitude value.

There are two significant differences between this sequence and the similar set of operations described in F.1.2 for Huffman coding. First, the sign is encoded before the magnitude category is identified, and second, the magnitude is decremented by 1 before the magnitude category is identified.

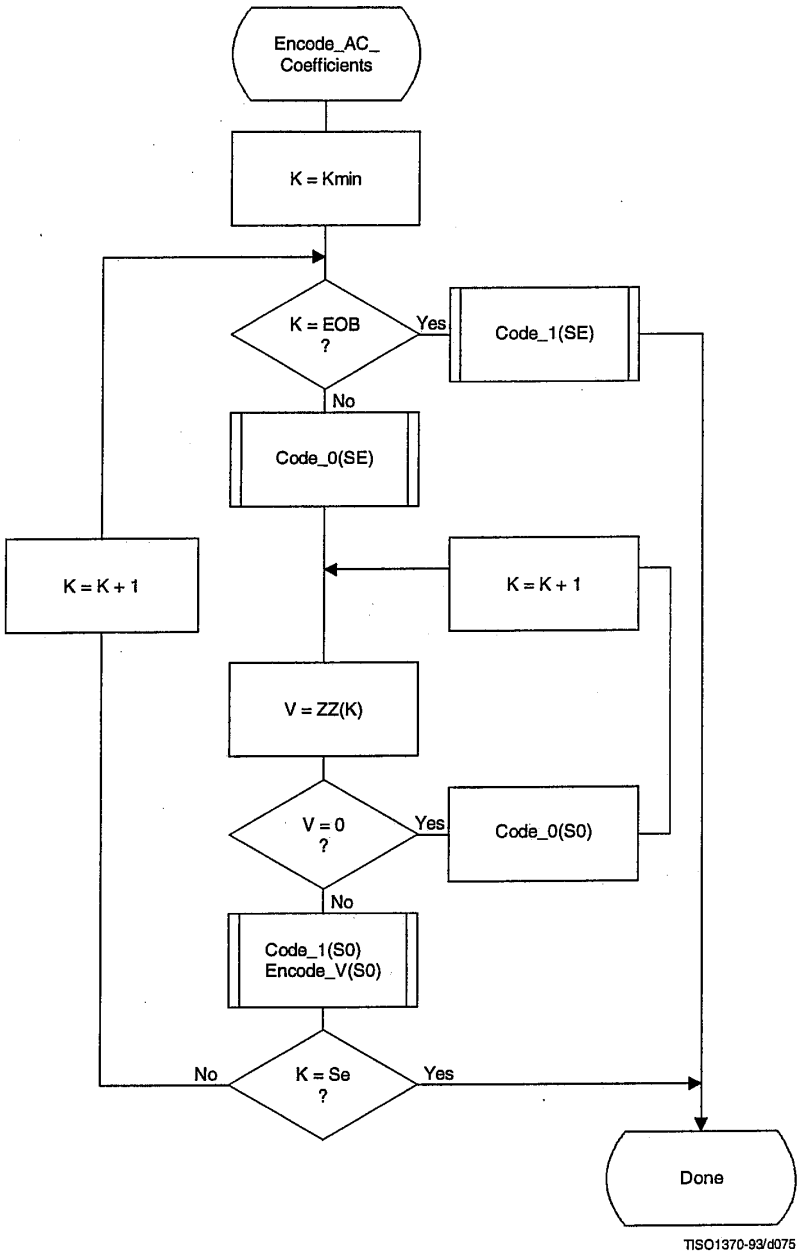


Figure F.5 – AC coding model for arithmetic coding

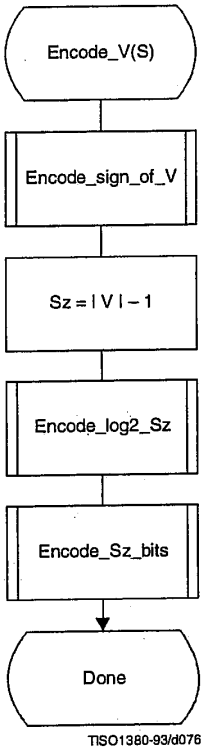


Figure F.6 – Sequence of procedures in encoding non-zero values of V

F.1.4.3.1.1 Encoding the sign

The sign is encoded by coding a 0-decision when the sign is positive and a 1-decision when the sign is negative (see Figure F.7).

The context-indices SS, SN and SP are defined for DC coding in Table F.4 and for AC coding in Table F.5. After the sign is coded, the context-index S is set to either SN or SP, establishing an initial value for Encode_log2_Sz.

F.1.4.3.1.2 Encoding the magnitude category

The magnitude category is determined by a sequence of binary decisions which compares Sz against an exponentially increasing bound (which is a power of 2) in order to determine the position of the leading 1-bit. This establishes the magnitude category in much the same way that the Huffman encoder generates a code for the value associated with the difference category. The flow chart for this procedure is shown in Figure F.8.

The starting value of the context-index S is determined in Encode_sign_of_V, and the context-index values X1 and X2 are defined for DC coding in Table F.4 and for AC coding in Table F.5. In Figure F.8, M is the exclusive upper bound for the magnitude and the abbreviations "SLL" and "SRL" refer to the shift-left-logical and shift-right-logical operations – in this case by one bit position. The SRL operation at the completion of the procedure aligns M with the most significant bit of Sz (see Table F.3).

The highest precision allowed for the DCT is 15 bits. Therefore, the highest precision required for the coding decision tree is 16 bits for the DC coefficient difference and 15 bits for the AC coefficients, including the sign bit.

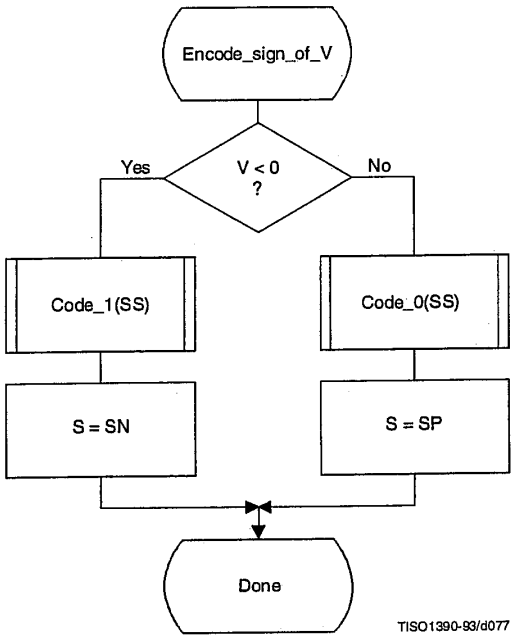


Figure F.7 – Encoding the sign of V

Table F.3 – Categories for each maximum bound

Exclusive upper bound (M)	Sz range	Number of low order magnitude bits
1	0	0
2	1	0
4	2,3	1
8	4,...,7	2
16	8,...,15	3
32	16,...,31	4
64	32,...,63	5
128	64,...,127	6
256	128,...,255	7
512	256,...,511	8
1 024	512,...,1 023	9
2 048	1 024,...,2 047	10
4 096	2 048,...,4 095	11
8 192	4 096,...,8 191	12
16 384	8 192,...,16 383	13
32 768	16 384,...,32 767	14

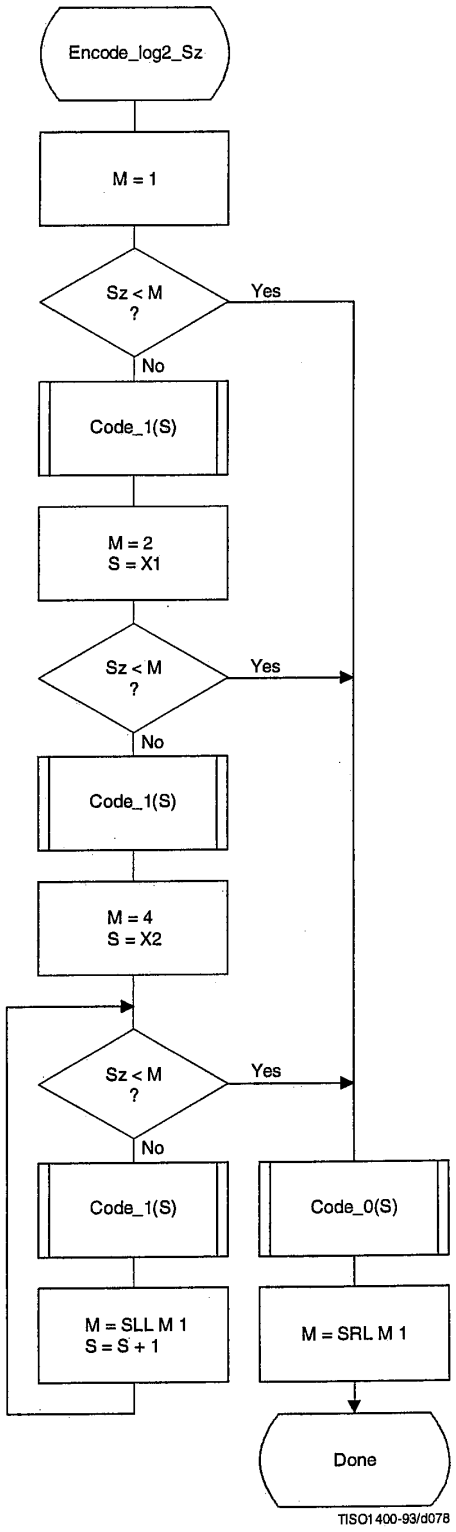


Figure F.8 – Decision sequence to establish the magnitude category

F.1.4.3.1.3 Encoding the exact value of the magnitude

After the magnitude category is encoded, the low order magnitude bits are encoded. These bits are encoded in order of decreasing bit significance. The procedure is shown in Figure F.9. The abbreviation "SRL" indicates the shift-right-logical operation, and M is the exclusive bound established in Figure F.8. Note that M has only one bit set – shifting M right converts it into a bit mask for the logical "AND" operation.

The starting value of the context-index S is determined in Encode_log2_Sz. The increment of S by 14 at the beginning of this procedure sets the context-index to the value required in Tables F.4 and F.5.

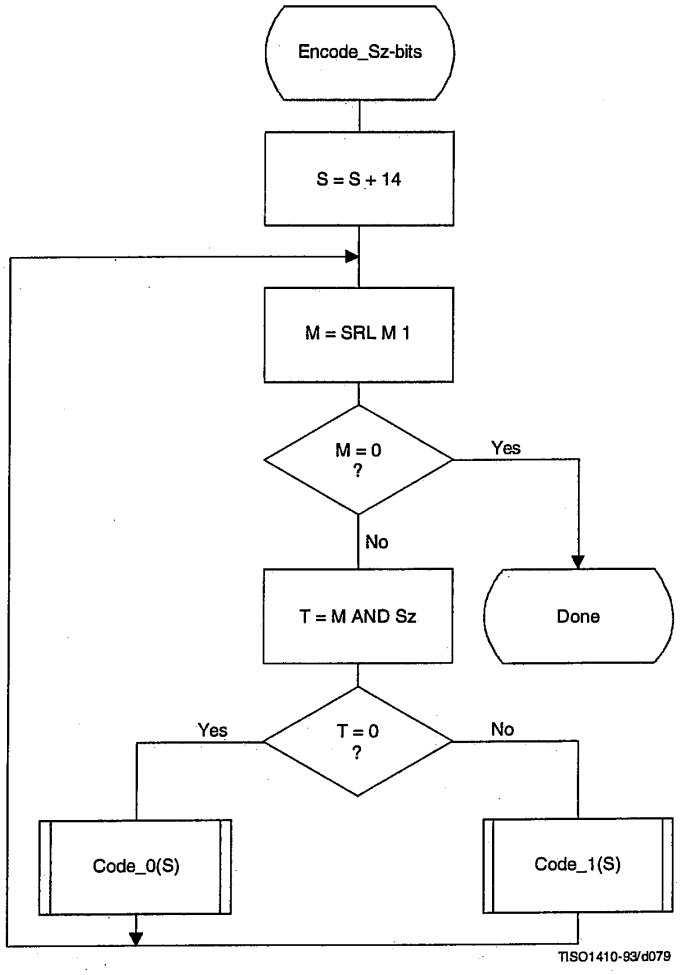


Figure F.9 – Decision sequence to code the magnitude bit pattern

F.1.4.4 Statistical models

An adaptive binary arithmetic coder requires a statistical model. The statistical model defines the contexts which are used to select the conditional probability estimates used in the encoding and decoding procedures.

Each decision in the binary decision trees is associated with one or more contexts. These contexts identify the sense of the MPS and the index in Table D.3 of the conditional probability estimate Qe which is used to encode and decode the binary decision.

The arithmetic coder is adaptive, which means that the probability estimates for each context are developed and maintained by the arithmetic coding system on the basis of prior coding decisions for that context.

F.1.4.4.1 Statistical model for coding DC prediction differences

The statistical model for coding the DC difference conditions some of the probability estimates for the binary decisions on previous DC coding decisions.

F.1.4.4.1.1 Statistical conditioning on sign

In coding the DC coefficients, four separate statistics bins (probability estimates) are used in coding the zero/not-zero ($V = 0$) decision, the sign decision and the first magnitude category decision. Two of these bins are used to code the $V = 0$ decision and the sign decision. The other two bins are used in coding the first magnitude decision, $Sz < 1$; one of these bins is used when the sign is positive, and the other is used when the sign is negative. Thus, the first magnitude decision probability estimate is conditioned on the sign of V.

F.1.4.4.1.2 Statistical conditioning on DC difference in previous block

The probability estimates for these first three decisions are also conditioned on Da, the difference value coded for the previous DCT block of the same component. The differences are classified into five groups: zero, small positive, small negative, large positive and large negative. The relationship between the default classification and the quantization scale is shown in Figure F.10.

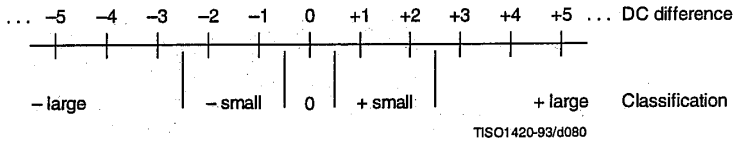


Figure F.10 – Conditioning classification of difference values

The bounds for the "small" difference category determine the classification. Defining L and U as integers in the range 0 to 15 inclusive, the lower bound (exclusive) for difference magnitudes classified as "small" is zero for $L = 0$, and is 2^{L-1} for $L > 0$.

The upper bound (inclusive) for difference magnitudes classified as "small" is 2^U.

L shall be less than or equal to U.

These bounds for the conditioning category provide a segmentation which is identical to that listed in Table F.3.

F.1.4.4.1.3 Assignment of statistical bins to the DC binary decision tree

As shown in Table F.4, each statistics area for DC coding consists of a set of 49 statistics bins. In the following explanation, it is assumed that the bins are contiguous. The first 20 bins consist of five sets of four bins selected by a context-index S0. The value of S0 is given by DC_Context(Da), which provides a value of 0, 4, 8, 12 or 16, depending on the difference classification of Da (see F.1.4.4.1.2). The remaining 29 bins, X1,...,X15,M2,...,M15, are used to code magnitude category decisions and magnitude bits.

Table F.4 – Statistical model for DC coefficient coding

Context-index	Value	Coding decision
S0	DC_Context(Da)	V = 0
SS	S0 + 1	Sign of V
SP	S0 + 2	Sz < 1 if V > 0
SN	S0 + 3	Sz < 1 if V < 0
X1	20	Sz < 2
X2	X1 + 1	Sz < 4
X3	X1 + 2	Sz < 8
.	.	.
.	.	.
X15	X1 + 14	$Sz < 2^{15}$
M2	X2 + 14	Magnitude bits if Sz < 4
M3	X3 + 14	Magnitude bits if Sz < 8
.	.	.
.	.	.
M15	X15 + 14	Magnitude bits if $Sz < 2^{15}$

F.1.4.4.1.4 Default conditioning for DC statistical model

The bounds, L and U, for determining the conditioning category have the default values L = 0 and U = 1. Other bounds may be set using the DAC (Define Arithmetic coding Conditioning) marker segment, as described in Annex B.

F.1.4.4.1.5 Initial conditions for DC statistical model

At the start of a scan and at the beginning of each restart interval, the difference for the previous DC value is defined to be zero in determining the conditioning state.

F.1.4.4.2 Statistical model for coding the AC coefficients

As shown in Table F.5, each statistics area for AC coding consists of a contiguous set of 245 statistics bins. Three bins are used for each value of the zig-zag index K, and two sets of 28 additional bins X2,...,X15,M2,...,M15 are used for coding the magnitude category and magnitude bits.

The value of SE (and also S0, SP and SN) is determined by the zig-zag index K. Since K is in the range 1 to 63, the lowest value for SE is 0 and the largest value for SP is 188. SS is not assigned a value in AC coefficient coding, as the signs of the coefficients are coded with a fixed probability value of approximately 0.5 (Qe = X'5A1D', MPS = 0).

The value of X2 is given by AC_Context(K). This gives X2 = 189 when $K \leq Kx$ and X2 = 217 when K > Kx, where Kx is defined using the DAC marker segment (see B.2.4.3).

Note that a X1 statistics bin is not used in this sequence. Instead, the 63 × 1 array of statistics bins for the magnitude category is used for two decisions. Once the magnitude bound has been determined – at statistics bin Xn, for example – a single statistics bin, Mn, is used to code the magnitude bit sequence for that bound.

F.1.4.4.2.1 Default conditioning for AC coefficient coding

The default value of Kx is 5. This may be modified using the DAC marker segment, as described in Annex B.

F.1.4.4.2.2 Initial conditions for AC statistical model

At the start of a scan and at each restart, all statistics bins are re-initialized to the standard default value described in Annex D.

Table F.5 – Statistical model for AC coefficient coding

Context-index	Value	Coding decision
SE	$3 \times (K - 1)$	K = EOB
S0	SE + 1	V = 0
SS	Fixed estimate	Sign of V
SN,SP	S0 + 1	Sz < 1
X1	S0 + 1	Sz < 2
X2	AC_Context(K)	Sz < 4
X3	X2 + 1	Sz < 8
.	.	.
.	.	.
X15	X2 + 13	$Sz < 2^{15}$
M2	X2 + 14	Magnitude bits if Sz < 4
M3	X3 + 14	Magnitude bits if Sz < 8
.	.	.
.	.	.
M15	X15 + 14	Magnitude bits if $Sz < 2^{15}$

F.1.5 Extended sequential DCT-based Huffman encoding process for 12-bit sample precision

This process is identical to the sequential DCT process for 8-bit precision extended to four Huffman table destinations as documented in F.1.3, with the following changes.

F.1.5.1 Structure of DC code table for 12-bit sample precision

The two's complement difference magnitudes are grouped into 16 categories, SSSS, and a Huffman code is created for each of the 16 difference magnitude categories.

The Huffman table for DC coding (see Table F.1) is extended as shown in Table F.6.

Table F.6 – Difference magnitude categories for DC coding

SSSS	Difference values
12	–4 095..–2 048,2 048..4 095
13	–8 191..–4 096,4 096..8 191
14	–16 383..–8 192,8 192..16 383
15	–32 767..–16 384,16 384..32 767

F.1.5.2 Structure of AC code table for 12-bit sample precision

The general structure of the code table is extended as illustrated in Figure F.11. The Huffman table for AC coding is extended as shown in Table F.7.

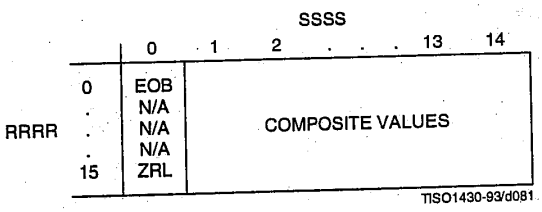


Figure F.11 – Two-dimensional value array for Huffman coding

Table F.7 – Values assigned to coefficient amplitude ranges

SSSS	AC coefficients
11	–2 047..–1 024,1 024..2 047
12	–4 095..–2 048,2 048..4 095
13	–8 191..–4 096,4 096..8 191
14	–16 383..–8 192,8 192..16 383

F.1.6 Extended sequential DCT-based arithmetic encoding process for 12-bit sample precision

The process is identical to the sequential DCT process for 8-bit precision except for changes in the precision of the FDCT computation.

The structure of the encoding procedure is identical to that specified in F.1.4 which was already defined for a 12-bit sample precision.

F.2 Sequential DCT-based decoding processes

F.2.1 Sequential DCT-based control procedures and coding models

F.2.1.1 Control procedures for sequential DCT-based decoders

The control procedures for decoding compressed image data and its constituent parts – the frame, scan, restart interval and MCU – are given in Figures E.6 to E.10. The procedure for decoding a MCU (Figure E.10) repetitively calls the procedure for decoding a data unit. For DCT-based decoders the data unit is an 8×8 block of samples.

F.2.1.2 Procedure for decoding an 8×8 block data unit

In the sequential DCT-based decoding process, decoding an 8×8 block data unit consists of the following procedures:

a) decode DC coefficient for 8×8 block using the DC table destination specified in the scan header;

b) decode AC coefficients for 8×8 block using the AC table destination specified in the scan header;

c) dequantize using table destination specified in the frame header and calculate the inverse 8×8 DCT.

F.2.1.3 Decoding models for the sequential DCT procedures

Two decoding procedures are used, one for the DC coefficient ZZ(0) and the other for the AC coefficients ZZ(1)...ZZ(63). The coefficients are decoded in the order in which they occur in the zig-zag sequence order, starting with the DC coefficient. The coefficients are represented as two's complement integers.

F.2.1.3.1 Decoding model for DC coefficients

The decoded difference, DIFF, is added to PRED, the DC value from the most recently decoded 8×8 block from the same component. Thus ZZ(0) = PRED + DIFF.

At the beginning of the scan and at the beginning of each restart interval, the prediction for the DC coefficient is initialized to zero.

F.2.1.3.2 Decoding model for AC coefficients

The AC coefficients are decoded in the order in which they occur in ZZ. When the EOB is decoded, all remaining coefficients in ZZ are initialized to zero.

F.2.1.4 Dequantization of the quantized DCT coefficients

The dequantization of the quantized DCT coefficients as described in Annex A, is accomplished by multiplying each quantized coefficient value by the quantization table value for that coefficient. The decoder shall be able to use up to four quantization table destinations.

F.2.1.5 Inverse DCT (IDCT)

The mathematical definition of the IDCT is given in A.3.3.

After computation of the IDCT, the signed output samples are level-shifted, as described in Annex A, converting the output to an unsigned representation. For 8-bit precision the level shift is performed by adding 128. For 12-bit precision the level shift is performed by adding 2 048. If necessary, the output samples shall be clamped to stay within the range appropriate for the precision (0 to 255 for 8-bit precision and 0 to 4 095 for 12-bit precision).

F.2.2 Baseline Huffman Decoding procedures

The baseline decoding procedure is for 8-bit sample precision. The decoder shall be capable of using up to two DC and two AC Huffman tables within one scan.

F.2.2.1 Huffman decoding of DC coefficients

The decoding procedure for the DC difference, DIFF, is:

T = DECODE

DIFF = RECEIVE(T)

DIFF = EXTEND(DIFF,T)

where DECODE is a procedure which returns the 8-bit value associated with the next Huffman code in the compressed image data (see F.2.2.3) and RECEIVE(T) is a procedure which places the next T bits of the serial bit string into the low order bits of DIFF, MSB first. If T is zero, DIFF is set to zero. EXTEND is a procedure which converts the partially decoded DIFF value of precision T to the full precision difference. EXTEND is shown in Figure F.12.

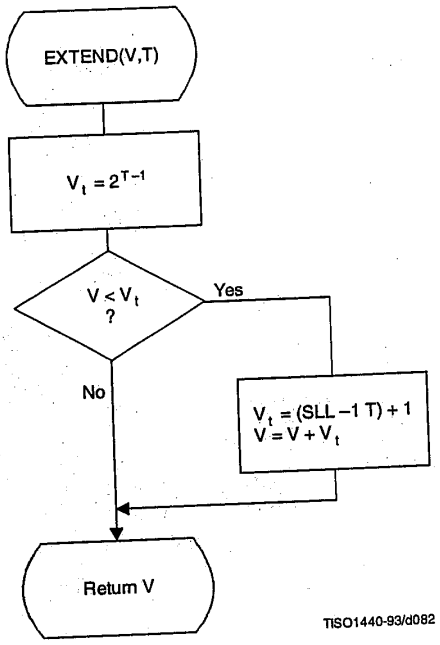


Figure F.12 – Extending the sign bit of a decoded value in V

F.2.2.2 Decoding procedure for AC coefficients

The decoding procedure for AC coefficients is shown in Figures F.13 and F.14.

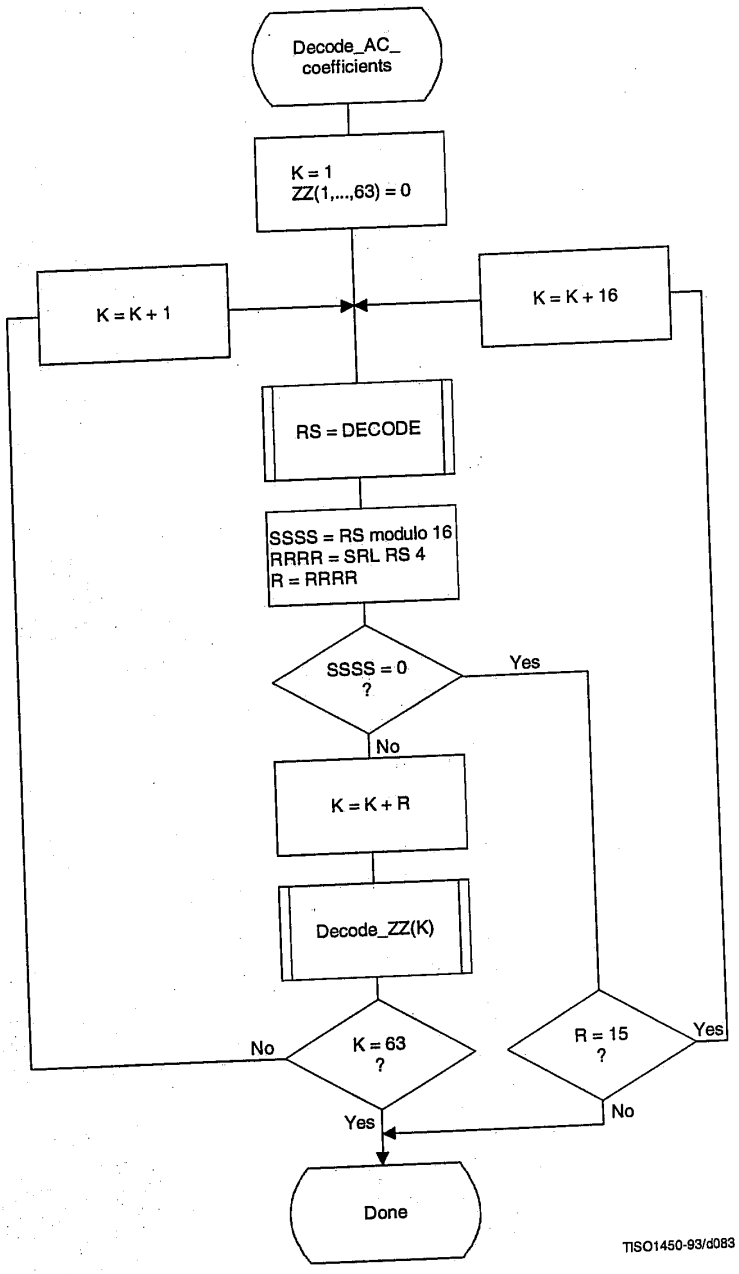


Figure F.13 – Huffman decoding procedure for AC coefficients

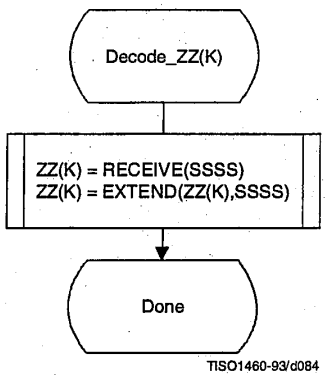


Figure F.14 – Decoding a non-zero AC coefficient

The decoding of the amplitude and sign of the non-zero coefficient is done in the procedure "Decode_ZZ(K)", shown in Figure F.14.

DECODE is a procedure which returns the value, RS, associated with the next Huffman code in the code stream (see F.2.2.3). The values SSSS and R are derived from RS. The value of SSSS is the four low order bits of the composite value and R contains the value of RRRR (the four high order bits of the composite value). The interpretation of these values is described in F.1.2.2. EXTEND is shown in Figure F.12.

F.2.2.3 The DECODE procedure

The DECODE procedure decodes an 8-bit value which, for the DC coefficient, determines the difference magnitude category. For the AC coefficient this 8-bit value determines the zero run length and non-zero coefficient category.

Three tables, HUFFVAL, HUFFCODE, and HUFFSIZE, have been defined in Annex C. This particular implementation of DECODE makes use of the ordering of the Huffman codes in HUFFCODE according to both value and code size. Many other implementations of DECODE are possible.

NOTE – The values in HUFFVAL are assigned to each code in HUFFCODE and HUFFSIZE in sequence. There are no ordering requirements for the values in HUFFVAL which have assigned codes of the same length.

The implementation of DECODE described in this subclause uses three tables, MINCODE, MAXCODE and VALPTR, to decode a pointer to the HUFFVAL table. MINCODE, MAXCODE and VALPTR each have 16 entries, one for each possible code size. MINCODE(I) contains the smallest code value for a given length I, MAXCODE(I) contains the largest code value for a given length I, and VALPTR(I) contains the index to the start of the list of values in HUFFVAL which are decoded by code words of length I. The values in MINCODE and MAXCODE are signed 16-bit integers; therefore, a value of –1 sets all of the bits.

The procedure for generating these tables is shown in Figure F.15. The procedure for DECODE is shown in Figure F.16. Note that the 8-bit "VALUE" is returned to the procedure which invokes DECODE.

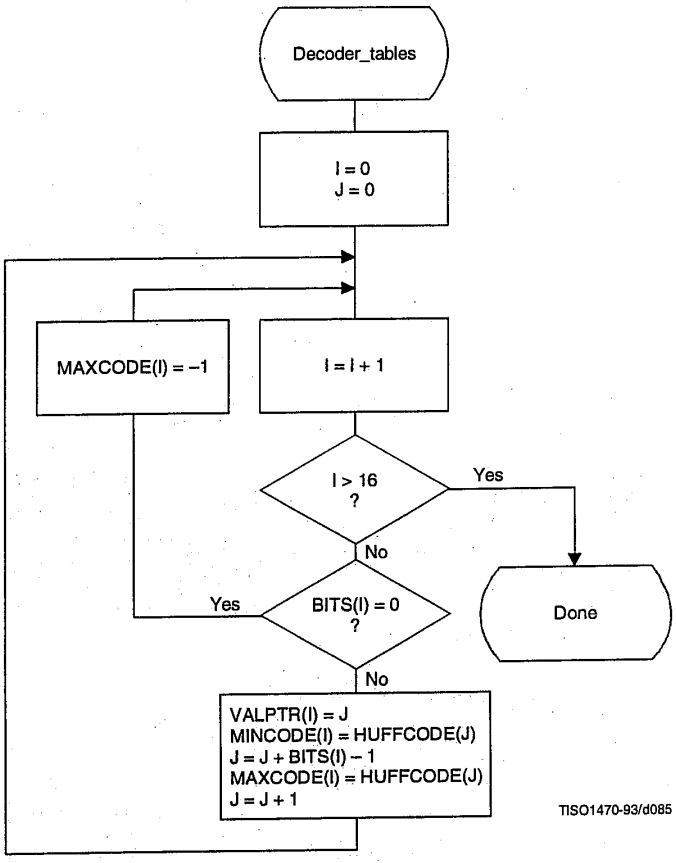


Figure F.15 – Decoder table generation

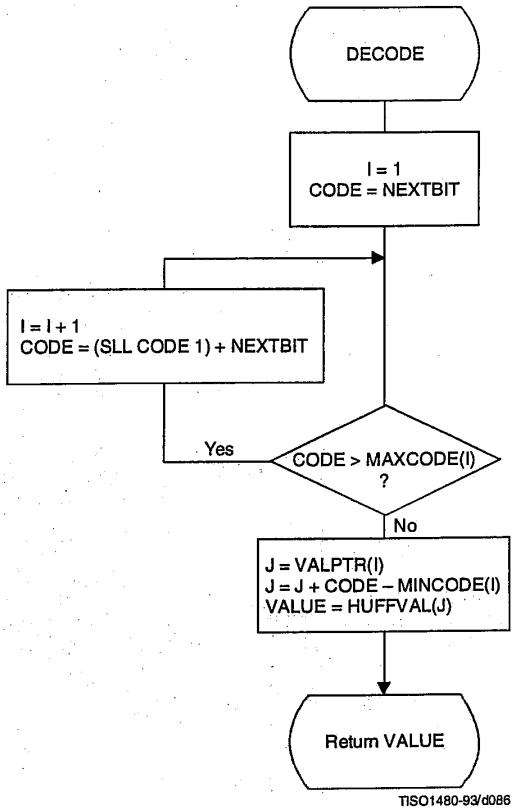


Figure F.16 – Procedure for DECODE

F.2.2.4 The RECEIVE procedure

RECEIVE(SSSS) is a procedure which places the next SSSS bits of the entropy-coded segment into the low order bits of DIFF, MSB first. It calls NEXTBIT and it returns the value of DIFF to the calling procedure (see Figure F.17).

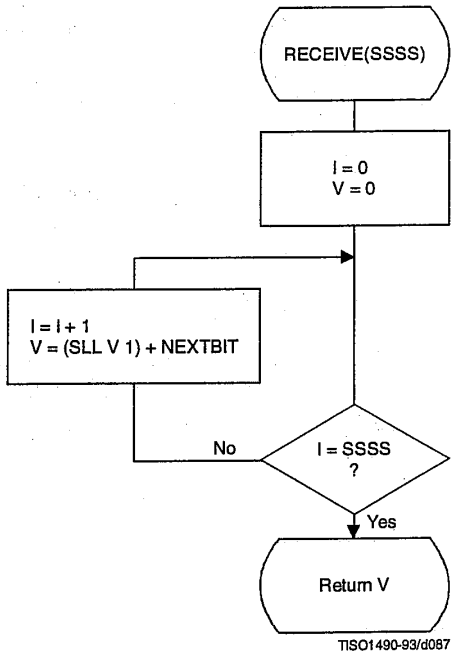


Figure F.17 – Procedure for RECEIVE(SSSS)

F.2.2.5 The NEXTBIT procedure

NEXTBIT reads the next bit of compressed data and passes it to higher level routines. It also intercepts and removes stuff bytes and detects markers. NEXTBIT reads the bits of a byte starting with the MSB (see Figure F.18).

Before starting the decoding of a scan, and after processing a RST marker, CNT is cleared. The compressed data are read one byte at a time, using the procedure NEXTBYTE. Each time a byte, B, is read, CNT is set to 8.

The only valid marker which may occur within the Huffman coded data is the RST_m marker. Other than the EOI or markers which may occur at or before the start of a scan, the only marker which can occur at the end of the scan is the DNL (define-number-of-lines).

Normally, the decoder will terminate the decoding at the end of the final restart interval before the terminating marker is intercepted. If the DNL marker is encountered, the current line count is set to the value specified by that marker. Since the DNL marker can only be used at the end of the first scan, the scan decode procedure must be terminated when it is encountered.

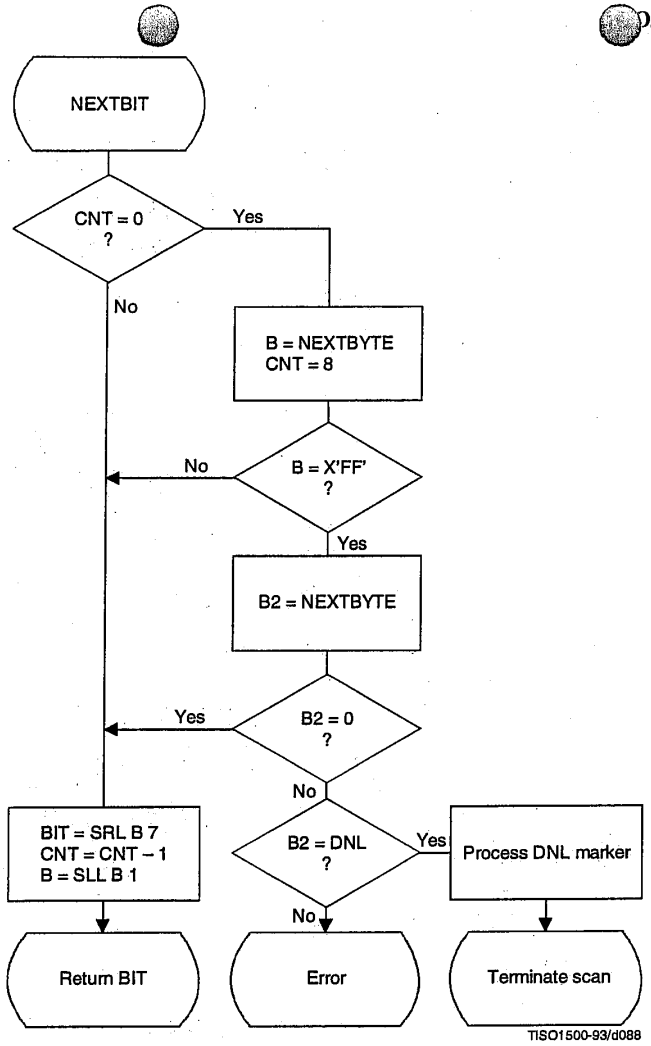


Figure F.18 – Procedure for fetching the next bit of compressed data

F.2.3 Sequential DCT decoding process with 8-bit precision extended to four sets of Huffman tables

This process is identical to the Baseline decoding process described in F.2.2, with the exception that the decoder shall be capable of using up to four DC and four AC Huffman tables within one scan. Four DC and four AC Huffman tables is the maximum allowed by this Specification.

F.2.4 Sequential DCT decoding process with arithmetic coding

This subclause describes the sequential DCT decoding process with arithmetic decoding.

The arithmetic decoding procedures for decoding binary decisions, initializing the statistical model, initializing the decoder, and resynchronizing the decoder are listed in Table D.4 of Annex D.

Some of the procedures in Table D.4 are used in the higher level control structure for scans and restart intervals described in F.2. At the beginning of scans and restart intervals, the probability estimates used in the arithmetic decoder are reset to the standard initial value as part of the Initdec procedure which restarts the arithmetic coder.

The statistical models defined in F.1.4.4 also apply to this decoding process.

The decoder shall be capable of using up to four DC and four AC conditioning tables and associated statistics areas within one scan.

F.2.4.1 Arithmetic decoding of DC coefficients

The basic structure of the decision sequence for decoding a DC difference value, DIFF, is shown in Figure F.19. The equivalent structure for the encoder is found in Figure F.4.

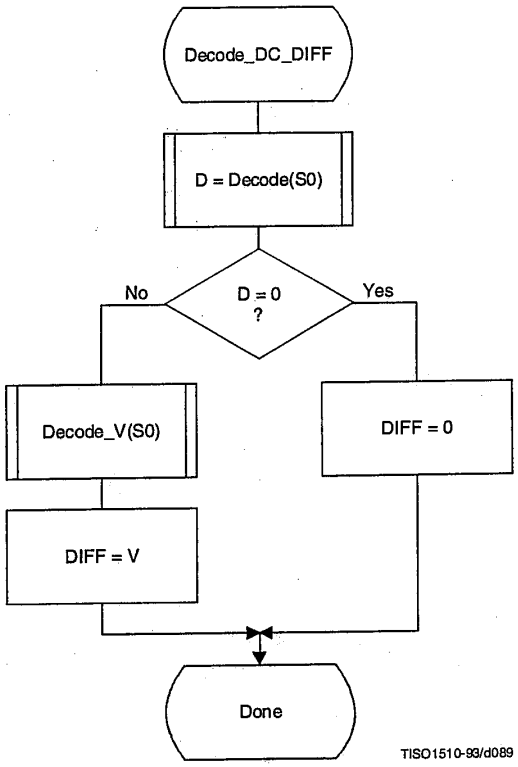


Figure F.19 – Arithmetic decoding of DC difference

The context-indices used in the DC decoding procedures are defined in Table F.4 (see F.1.4.4.1.3).

The "Decode" procedure returns the value "D" of the binary decision. If the value is not zero, the sign and magnitude of the non-zero DIFF must be decoded by the procedure "Decode_V(S0)".

F.2.4.2 Arithmetic Decoding of AC coefficients

The AC coefficients are decoded in the order that they occur in ZZ(1,...,63). The encoder procedure for the coding process is found in Figure F.5. Figure F.20 illustrates the decoding sequence.

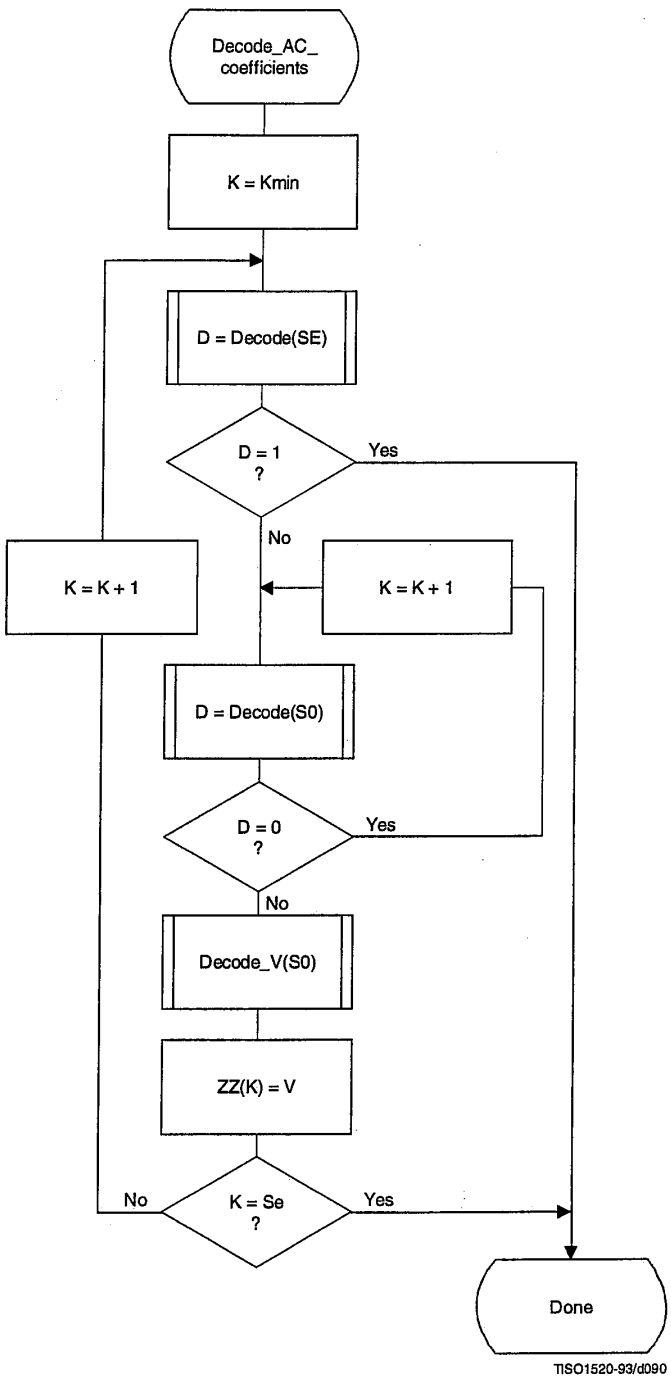


Figure F.20 – Procedure for decoding the AC coefficients

The context-indices used in the AC decoding procedures are defined in Table F.5 (see F.1.4.4.2).

In Figure F.20, K is the index to the zig-zag sequence position. For the sequential scan, Kmin = 1 and Se = 63. The decision at the top of the loop is the EOB decision. If the EOB occurs (D = 1), the remaining coefficients in the block are set to zero. The inner loop just below the EOB decoding decodes runs of zero coefficients. Whenever the coefficient is non-zero, "Decode_V" decodes the sign and magnitude of the coefficient. After each non-zero coefficient is decoded, the EOB decision is again decoded unless K = Se.

F.2.4.3 Decoding the binary decision sequence for non-zero DC differences and AC coefficients

Both the DC difference and the AC coefficients are represented as signed two's complement 16-bit integer values. The decoding decision tree for these signed integer values is the same for both the DC and AC coding models. Note, however, that the statistical models are not the same.

F.2.4.3.1 Arithmetic decoding of non-zero values

Denoting either DC differences or AC coefficients as V, the non-zero signed integer value of V is decoded by the sequence shown in Figure F.21. This sequence first decodes the sign of V. It then decodes the magnitude category of V (Decode_log2_Sz), and then decodes the low order magnitude bits (Decode_Sz_bits). Note that the value decoded for Sz must be incremented by 1 to get the actual coefficient magnitude.

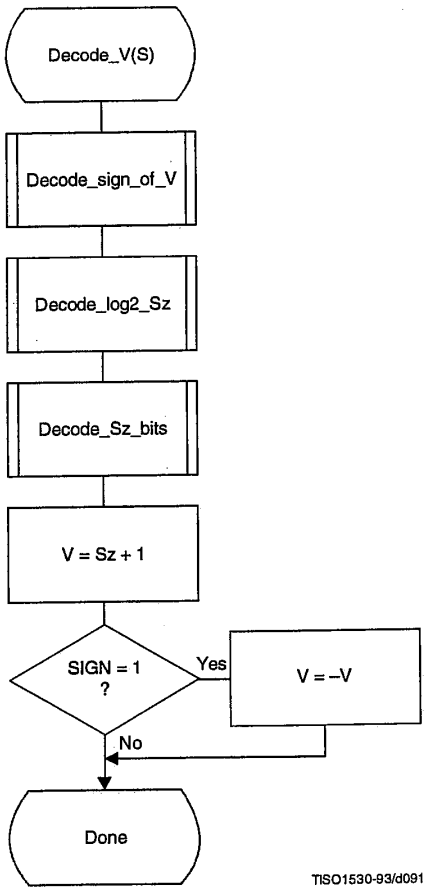


Figure F.21 – Sequence of procedures in decoding non-zero values of V

F.2.4.3.1.1 Decoding the sign

The sign is decoded by the procedure shown in Figure F.22.

The context-indices are defined for DC decoding in Table F.4 and AC decoding in Table F.5.

If SIGN = 0, the sign of the coefficient is positive; if SIGN = 1, the sign of the coefficient is negative.

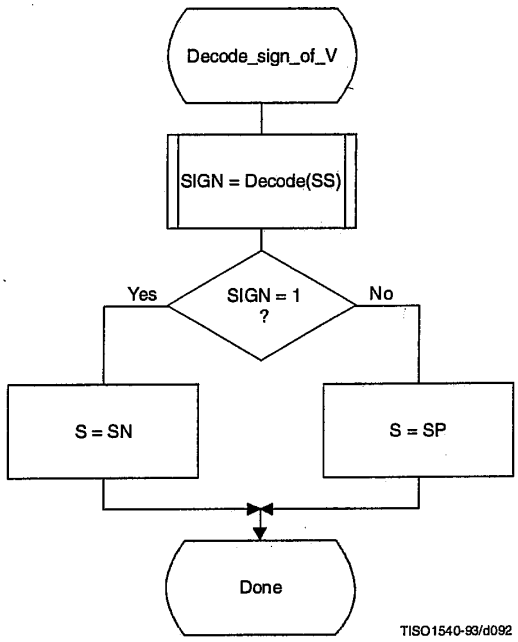


Figure F.22 – Decoding the sign of V

F.2.4.3.1.2 Decoding the magnitude category

The context-index S is set in Decode_sign_of_V and the context-index values X1 and X2 are defined for DC coding in Table F.4 and for AC coding in Table F.5.

In Figure F.23, M is set to the upper bound for the magnitude and shifted left until the decoded decision is zero. It is then shifted right by 1 to become the leading bit of the magnitude of Sz.

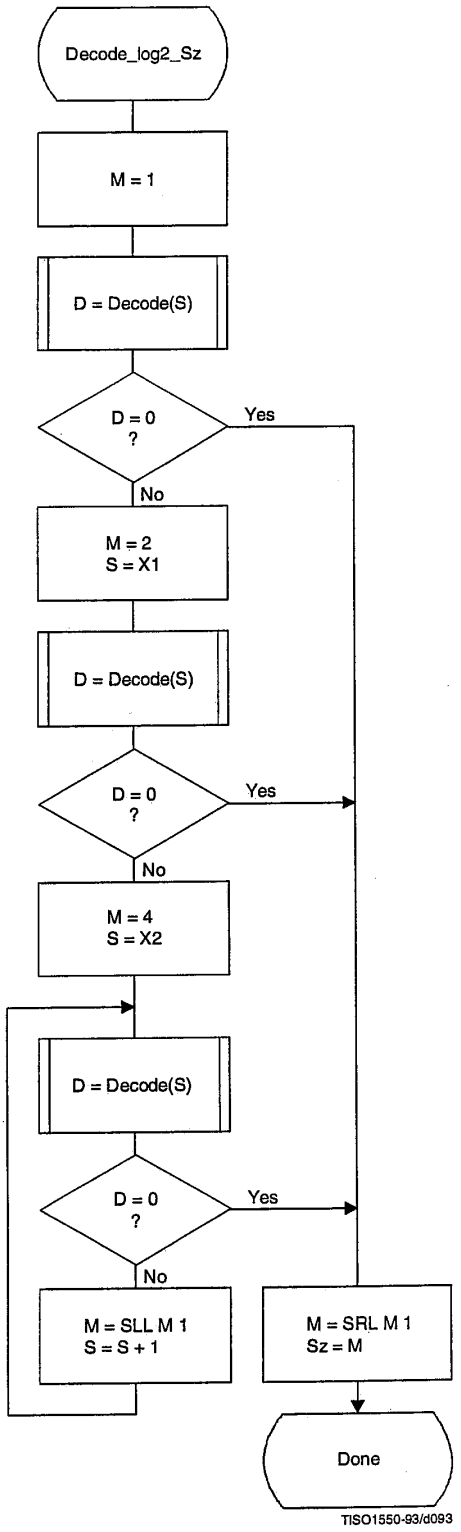


Figure F.23 – Decoding procedure to establish the magnitude category

F.2.4.3.1.3 Decoding the exact value of the magnitude

After the magnitude category is decoded, the low order magnitude bits are decoded. These bits are decoded in order of decreasing bit significance. The procedure is shown in Figure F.24.

The context-index S is set in Decode_log2_Sz.

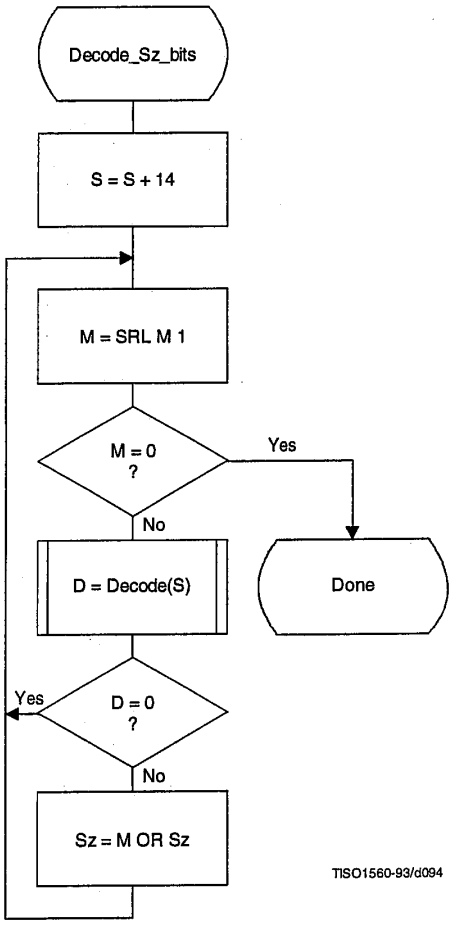


Figure F.24 – Decision sequence to decode the magnitude bit pattern

F.2.4.4 Decoder restart

The RST_m markers which are added to the compressed data between each restart interval have a two byte value which cannot be generated by the coding procedures. These two byte sequences can be located without decoding, and can therefore be used to resynchronize the decoder. RST_m markers can therefore be used for error recovery.

Before error recovery procedures can be invoked, the error condition must first be detected. Errors during decoding can show up in two places:

a) The decoder fails to find the expected marker at the point where it is expecting resynchronization.

b) Physically impossible data are decoded. For example, decoding a magnitude beyond the range of values allowed by the model is quite likely when the compressed data are corrupted by errors. For arithmetic decoders this error condition is extremely important to detect, as otherwise the decoder may reach a condition where it uses the compressed data very slowly.

NOTE – Some errors will not cause the decoder to lose synchronization. In addition, recovery is not possible for all errors; for example, errors in the headers are likely to be catastrophic. The two error conditions listed above, however, almost always cause the decoder to lose synchronization in a way which permits recovery.

In regaining synchronization, the decoder can make use of the modulo 8 coding restart interval number in the low order bits of the RST_m marker. By comparing the expected restart interval number to the value in the next RST_m marker in the compressed image data, the decoder can usually recover synchronization. It then fills in missing lines in the output data by replication or some other suitable procedure, and continues decoding. Of course, the reconstructed image will usually be highly corrupted for at least a part of the restart interval where the error occurred.

F.2.5 Sequential DCT decoding process with Huffman coding and 12-bit precision

This process is identical to the sequential DCT process defined for 8-bit sample precision and extended to four Huffman tables, as documented in F.2.3, but with the following changes.

F.2.5.1 Structure of DC Huffman decode table

The general structure of the DC Huffman decode table is extended as described in F.1.5.1.

F.2.5.2 Structure of AC Huffman decode table

The general structure of the AC Huffman decode table is extended as described in F.1.5.2.

F.2.6 Sequential DCT decoding process with arithmetic coding and 12-bit precision

The process is identical to the sequential DCT process for 8-bit precision except for changes in the precision of the IDCT computation.

The structure of the decoding procedure in F.2.4 is already defined for a 12-bit input precision.

Annex G

Progressive DCT-based mode of operation

(This annex forms an integral part of this Recommendation | International Standard)

This annex provides a **functional specification** of the following coding processes for the progressive DCT-based mode of operation:

1) spectral selection only, Huffman coding, 8-bit sample precision;
2) spectral selection only, arithmetic coding, 8-bit sample precision;
3) full progression, Huffman coding, 8-bit sample precision;
4) full progression, arithmetic coding, 8-bit sample precision;
5) spectral selection only, Huffman coding, 12-bit sample precision;
6) spectral selection only, arithmetic coding, 12-bit sample precision;
7) full progression, Huffman coding, 12-bit sample precision;
8) full progression, arithmetic coding, 12-bit sample precision.

For each of these, the encoding process is specified in G.1, and the decoding process is specified in G.2. The functional specification is presented by means of specific flow charts for the various procedures which comprise these coding processes.

NOTE – There is **no requirement** in this Specification that any encoder or decoder which embodies one of the above-named processes shall implement the procedures in precisely the manner specified by the flow charts in this annex. It is necessary only that an encoder or decoder implement the **function** specified in this annex. The sole criterion for an encoder or decoder to be considered in compliance with this Specification is that it satisfy the requirements given in clause 6 (for encoders) or clause 7 (for decoders), as determined by the compliance tests specified in Part 2.

The number of Huffman or arithmetic conditioning tables which may be used within the same scan is four.

Two complementary progressive procedures are defined, spectral selection and successive approximation.

In spectral selection the DCT coefficients of each block are segmented into frequency bands. The bands are coded in separate scans.

In successive approximation the DCT coefficients are divided by a power of two before coding. In the decoder the coefficients are multiplied by that same power of two before computing the IDCT. In the succeeding scans the precision of the coefficients is increased by one bit in each scan until full precision is reached.

An encoder or decoder implementing a full progression uses spectral selection within successive approximation. An allowed subset is spectral selection alone.

Figure G.1 illustrates the spectral selection and successive approximation progressive processes.

G.1 Progressive DCT-based encoding processes

G.1.1 Control procedures and coding models for progressive DCT-based procedures

G.1.1.1 Control procedures for progressive DCT-based encoders

The control procedures for encoding an image and its constituent parts – the frame, scan, restart interval and MCU – are given in Figures E.1 through E.5.

The control structure for encoding a frame is the same as for the sequential procedures. However, it is convenient to calculate the FDCT for the entire set of components in a frame before starting the scans. A buffer which is large enough to store all of the DCT coefficients may be used for this progressive mode of operation.

The number of scans is determined by the progression defined; the number of scans may be much larger than the number of components in the frame.

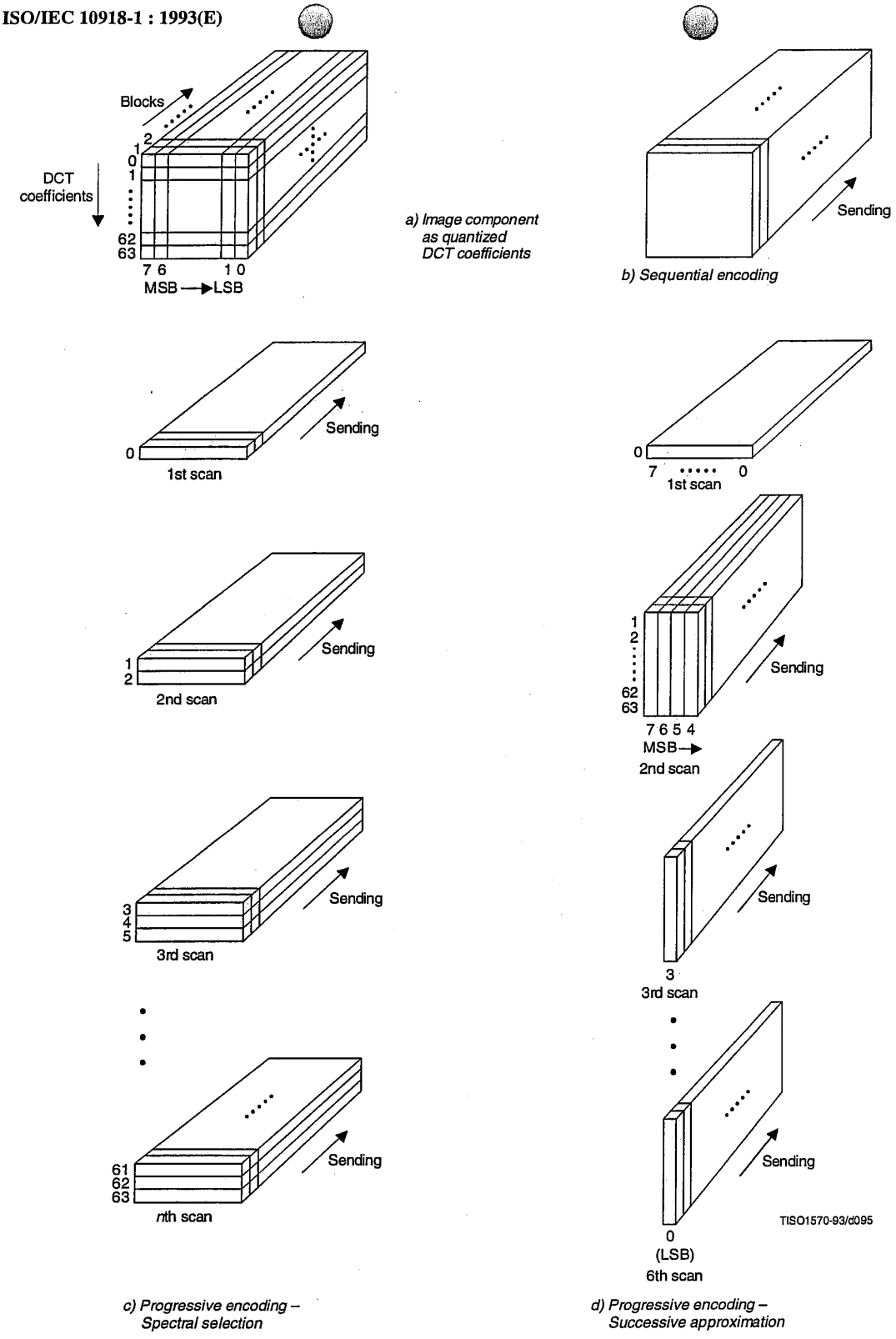


Figure G.1 – Spectral selection and successive approximation progressive processes

The procedure for encoding a MCU (see Figure E.5) repetitively invokes the procedure for coding a data unit. For DCT-based encoders the data unit is an 8 × 8 block of samples.

Only a portion of each 8 × 8 block is coded in each scan, the portion being determined by the scan header parameters Ss, Se, Ah, and Al (see B.2.3). The procedures used to code portions of each 8 × 8 block are described in this annex. Note, however, that where these procedures are identical to those used in the sequential DCT-based mode of operation, the sequential procedures are simply referenced.

G.1.1.1.1 Spectral selection control

In spectral selection the zig-zag sequence of DCT coefficients is segmented into bands. A band is defined in the scan header by specifying the starting and ending indices in the zig-zag sequence. One band is coded in a given scan of the progression. DC coefficients are always coded separately from AC coefficients, and only scans which code DC coefficients may have interleaved blocks from more than one component. All other scans shall have only one component. With the exception of the first DC scans for the components, the sequence of bands defined in the scans need not follow the zig-zag ordering. For each component, a first DC scan shall precede any AC scans.

G.1.1.1.2 Successive approximation control

If successive approximation is used, the DCT coefficients are reduced in precision by the point transform (see A.4) defined in the scan header (see B.2.3). The successive approximation bit position parameter Al specifies the actual point transform, and the high four bits (Ah) – if there are preceding scans for the band – contain the value of the point transform used in those preceding scans. If there are no preceding scans for the band, Ah is zero.

Each scan which follows the first scan for a given band progressively improves the precision of the coefficients by one bit, until full precision is reached.

G.1.1.2 Coding models for progressive DCT-based encoders

If successive approximation is used, the DCT coefficients are reduced in precision by the point transform (see A.4) defined in the scan header (see B.2.3). These models also apply to the progressive DCT-based encoders, but with the following changes.

G.1.1.2.1 Progressive encoding model for DC coefficients

If Al is not zero, the point transform for DC coefficients shall be used to reduce the precision of the DC coefficients. If Ah is zero, the coefficient values (as modified by the point transform) shall be coded, using the procedure described in Annex F. If Ah is not zero, the least significant bit of the point transformed DC coefficients shall be coded, using the procedures described in this annex.

G.1.1.2.2 Progressive encoding model for AC coefficients

If Al is not zero, the point transform for AC coefficients shall be used to reduce the precision of the AC coefficients. If Ah is zero, the coefficient values (as modified by the point transform) shall be coded using modifications of the procedures described in Annex F. These modifications are described in this annex. If Ah is not zero, the precision of the coefficients shall be improved using the procedures described in this annex.

G.1.2 Progressive encoding procedures with Huffman coding

G.1.2.1 Progressive encoding of DC coefficients with Huffman coding

The first scan for a given component shall encode the DC coefficient values using the procedures described in F.1.2.1. If the successive approximation bit position parameter Al is not zero, the coefficient values shall be reduced in precision by the point transform described in Annex A before coding.

In subsequent scans using successive approximation the least significant bits are appended to the compressed bit stream without compression or modification (see G.1.2.3), except for byte stuffing.

G.1.2.2 Progressive encoding of AC coefficients with Huffman coding

In spectral selection and in the first scan of successive approximation for a component, the AC coefficient coding model is similar to that used by the sequential procedures. However, the Huffman code tables are extended to include coding of runs of End-Of-Bands (EOBs). See Table G.1.

Table G.1 – EOBn code run length extensions

EOBn code	Run length
EOB0	1
EOB1	2,3
EOB2	4..7
EOB3	8..15
EOB4	16..31
EOB5	32..63
EOB6	64..127
EOB7	128..255
EOB8	256..511
EOB9	512..1 023
EOB10	1 024..2 047
EOB11	2 048..4 095
EOB12	4 096..8 191
EOB13	8 192..16 383
EOB14	16 384..32 767

The end-of-band run structure allows efficient coding of blocks which have only zero coefficients. An EOB run of length 5 means that the current block and the next four blocks have an end-of-band with no intervening non-zero coefficients. The EOB run length is limited only by the restart interval.

The extension of the code table is illustrated in Figure G.2.

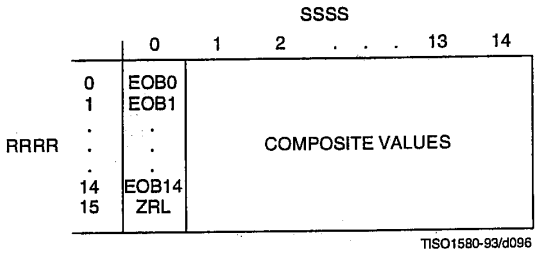


Figure G.2 – Two-dimensional value array for Huffman coding

The EOBn code sequence is defined as follows. Each EOBn code is followed by an extension field similar to the extension field for the coefficient amplitudes (but with positive numbers only). The number of bits appended to the EOBn code is the minimum number required to specify the run length.

If an EOB run is greater than 32 767, it is coded as a sequence of EOB runs of length 32 767 followed by a final EOB run sufficient to complete the run.

At the beginning of each restart interval the EOB run count, EOBRUN, is set to zero. At the end of each restart interval any remaining EOB run is coded.

The Huffman encoding procedure for AC coefficients in spectral selection and in the first scan of successive approximation is illustrated in Figures G.3, G.4, G.5, and G.6.

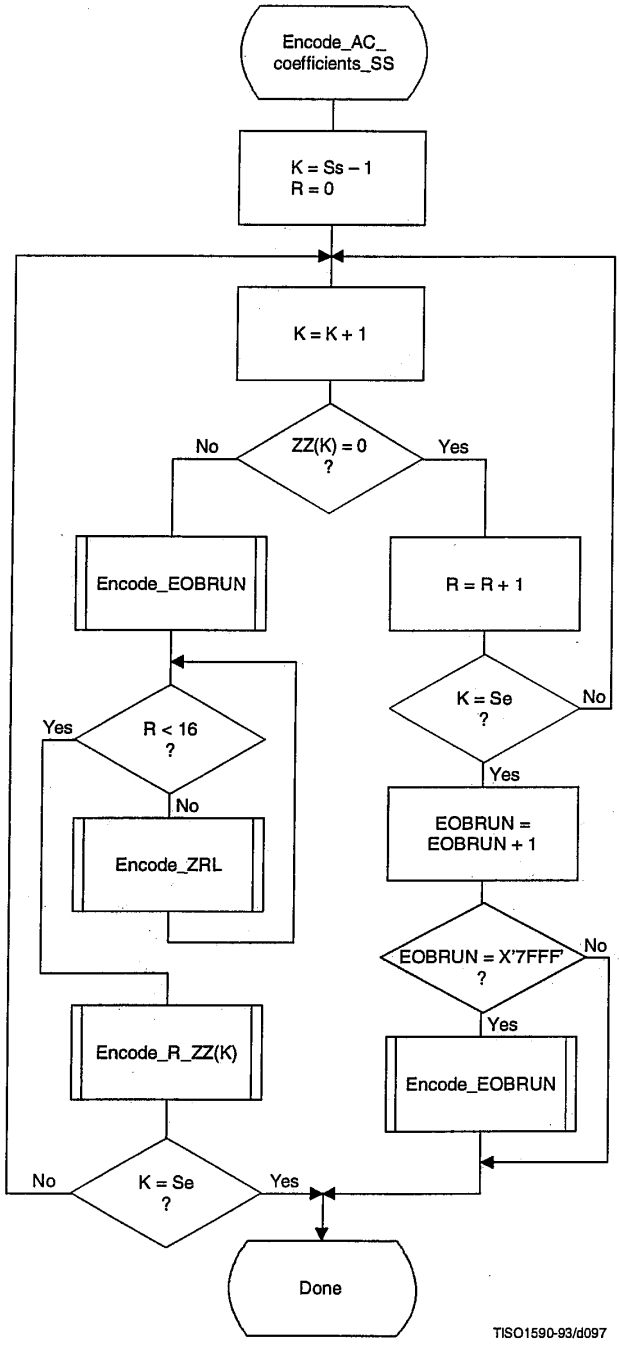


Figure G.3 – Procedure for progressive encoding of AC coefficients with Huffman coding

In Figure G.3, Ss is the start of spectral selection, Se is the end of spectral selection, K is the index into the list of coefficients stored in the zig-zag sequence ZZ, R is the run length of zero coefficients, and EOBRUN is the run length of EOBs. EOBRUN is set to zero at the start of each restart interval.

If the scan header parameter Al (successive approximation bit position low) is not zero, the DCT coefficient values ZZ(K) in Figure G.3 and figures which follow in this annex, including those in the arithmetic coding section, shall be replaced by the point transformed values ZZ'(K), where ZZ'(K) is defined by:

$$ZZ'(K) = \frac{ZZ(K)x}{2^{Al}}$$

EOBSIZE is a procedure which returns the size of the EOB extension field given the EOB run length as input. CSIZE is a procedure which maps an AC coefficient to the SSSS value defined in the subclauses on sequential encoding (see F.1.1 and F.1.3).

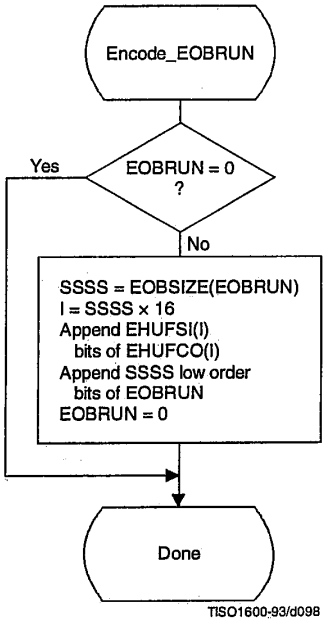


Figure G.4 – Progressive encoding of a non-zero AC coefficient

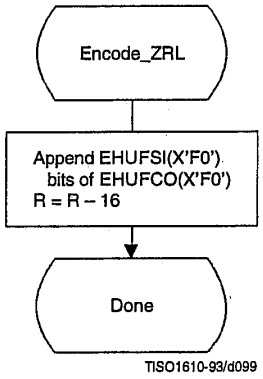


Figure G.5 – Encoding of the run of zero coefficients

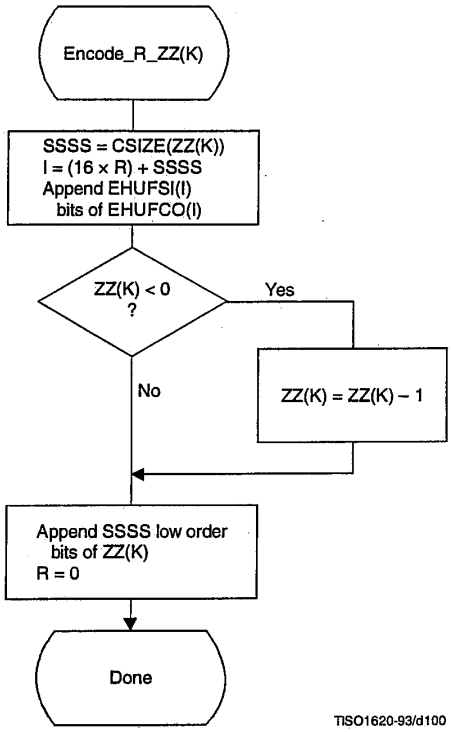


Figure G.6 – Encoding of the zero run and non-zero coefficient

G.1.2.3 Coding model for subsequent scans of successive approximation

The Huffman coding structure of the subsequent scans of successive approximation for a given component is similar to the coding structure of the first scan of that component.

The structure of the AC code table is identical to the structure described in G.1.2.2. Each non-zero point transformed coefficient that has a zero history (i.e. that has a value ± 1, and therefore has not been coded in a previous scan) is defined by a composite 8-bit run length-magnitude value of the form:

RRRRSSSS

The four most significant bits, RRRR, give the number of zero coefficients that are between the current coefficient and the previously coded coefficient (or the start of band). Coefficients with non-zero history (a non-zero value coded in a previous scan) are skipped over when counting the zero coefficients. The four least significant bits, SSSS, provide the magnitude category of the non-zero coefficient; for a given component the value of SSSS can only be one.

The run length-magnitude composite value is Huffman coded and each Huffman code is followed by additional bits:

a) One bit codes the sign of the newly non-zero coefficient. A 0-bit codes a negative sign; a 1-bit codes a positive sign.

b) For each coefficient with a non-zero history, one bit is used to code the correction. A 0-bit means no correction and a 1-bit means that one shall be added to the (scaled) decoded magnitude of the coefficient.

Non-zero coefficients with zero history are coded with a composite code of the form:

HUFFCO(RRRRSSSS) + additional bit (rule a) + correction bits (rule b)

In addition whenever zero runs are coded with ZRL or EOBn codes, correction bits for those coefficients with non-zero history contained within the zero run are appended according to rule b above.

For the Huffman coding version of Encode_AC_Coefficients_SA the EOB is defined to be the position of the last point transformed coefficient of magnitude 1 in the band. If there are no coefficients of magnitude 1, the EOB is defined to be zero.

NOTE – The definition of EOB is different for Huffman and arithmetic coding procedures.

In Figures G.7 and G.8 BE is the count of buffered correction bits at the start of coding of the block. BE is initialized to zero at the start of each restart interval. At the end of each restart interval any remaining buffered bits are appended to the bit stream following the last EOBn Huffman code and associated appended bits.

In Figures G.7 and G.9, BR is the count of buffered correction bits which are appended to the bit stream according to rule b. BR is set to zero at the beginning of each Encode_AC_Coefficients_SA. At the end of each restart interval any remaining buffered bits are appended to the bit stream following the last Huffman code and associated appended bits.

G.1.3 Progressive encoding procedures with arithmetic coding

G.1.3.1 Progressive encoding of DC coefficients with arithmetic coding

The first scan for a given component shall encode the DC coefficient values using the procedures described in F.1.4.1. If the successive approximation bit position parameter is not zero, the coefficient values shall be reduced in precision by the point transform described in Annex A before coding.

In subsequent scans using successive approximation the least significant bits shall be coded as binary decisions using a fixed probability estimate of 0.5 (Qe = X'5A1D', MPS = 0).

G.1.3.2 Progressive encoding of AC coefficients with arithmetic coding

Except for the point transform scaling of the DCT coefficients and the grouping of the coefficients into bands, the first scan(s) of successive approximation is identical to the sequential encoding procedure described in F.1.4. If Kmin is equated to Ss, the index of the first AC coefficient index in the band, the flow chart shown in Figure F.5 applies. The EOB decision in that figure refers to the "end-of-band" rather than the "end-of-block". For the arithmetic coding version of Encode_AC_Coefficients_SA (and all other AC coefficient coding procedures) the EOB is defined to be the position following the last non-zero coefficient in the band.

NOTE - The definition of EOB is different for Huffman and arithmetic coding procedures.

The statistical model described in F.1.4 also holds. For this model the default value of Kx is 5. Other values of Kx may be specified using the DAC marker code (Annex B). The following calculation for Kx has proven to give good results for 8-bit precision samples:

$$Kx = Kmin + SRL\ (8 + Se - Kmin)\ 4$$

This expression reduces to the default of Kx = 5 when the band is from index 1 to index 63.

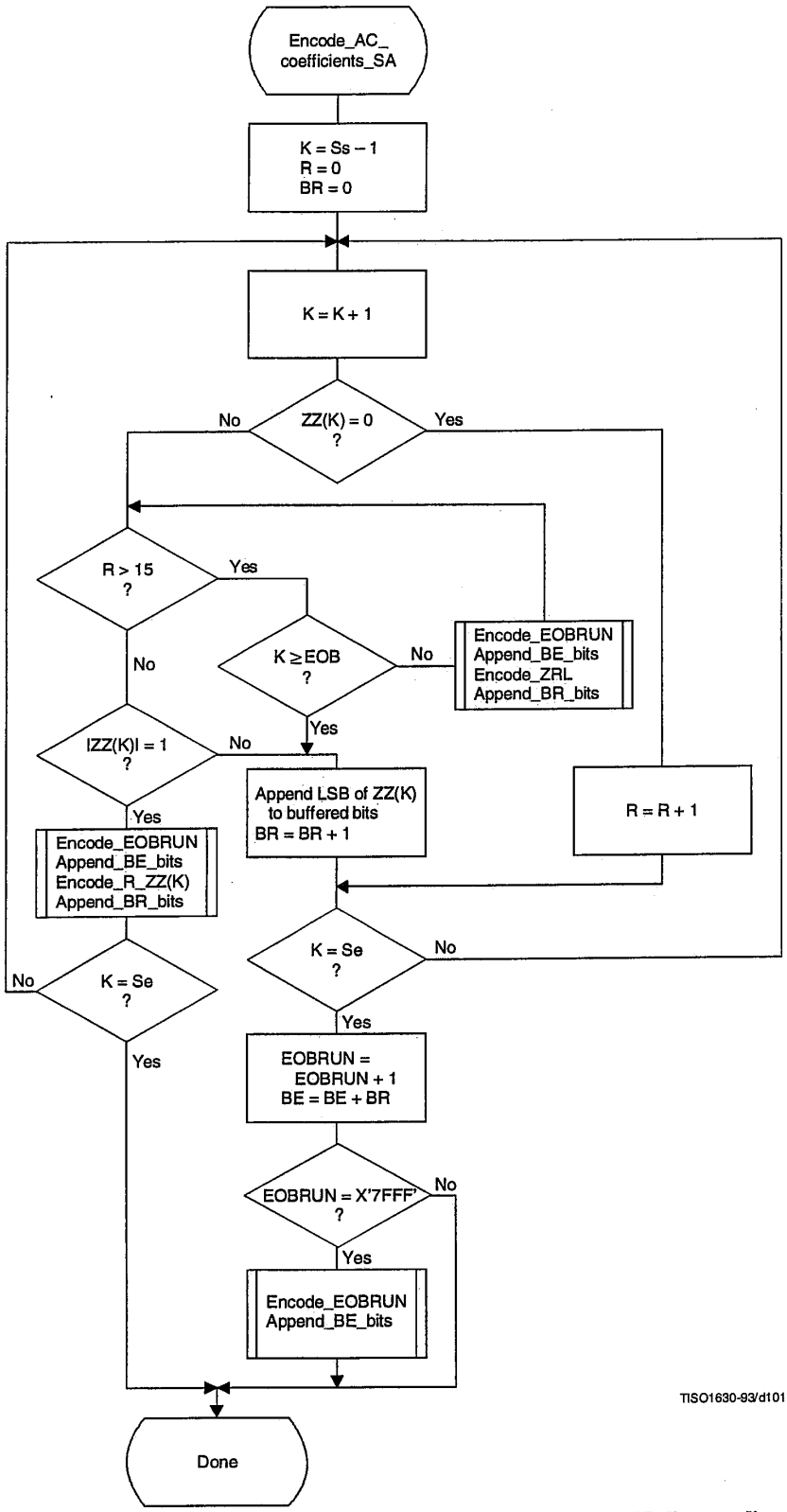


Figure G.7 – Successive approximation coding of AC coefficients using Huffman coding

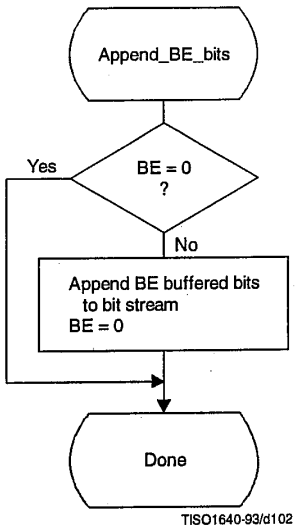


Figure G.8 – Transferring BE buffered bits from buffer to bit stream

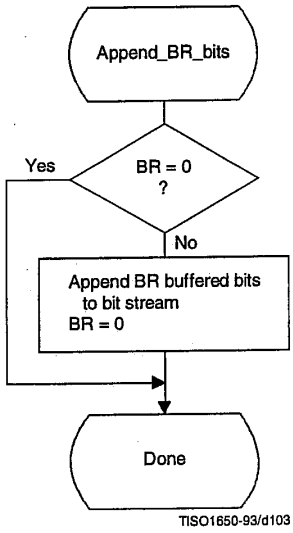


Figure G.9 – Transferring BR buffered bits from buffer to bit stream

G.1.3.3 Coding model for subsequent scans of successive approximation

The procedure "Encode_AC_Coefficient_SA" shown in Figure G.10 increases the precision of the AC coefficient values in the band by one bit.

As in the first scan of successive approximation for a component, an EOB decision is coded at the start of the band and after each non-zero coefficient.

However, since the end-of-band index of the previous successive approximation scan for a given component, EOBx, is known from the data coded in the prior scan of that component, this decision is bypassed whenever the current index, K, is less than EOBx. As in the first scan(s), the EOB decision is also bypassed whenever the last coefficient in the band is not zero. The decision ZZ(K) = 0 decodes runs of zero coefficients. If the decoder is at this step of the procedure, at least one non-zero coefficient remains in the band of the block being coded. If ZZ(K) is not zero, the procedure in Figure G.11 is followed to code the value.

The context-indices in Figures G.10 and G.11 are defined in Table G.2 (see G.1.3.3.1). The signs of coefficients with magnitude of one are coded with a fixed probability value of approximately 0.5 (Qe = X'5A1D', MPS = 0).

G.1.3.3.1 Statistical model for subsequent successive approximation scans

As shown in Table G.2, each statistics area for subsequent successive approximation scans of AC coefficients consists of a contiguous set of 189 statistics bins. The signs of coefficients with magnitude of one are coded with a fixed probability value of approximately 0.5 (Qe = X'5A1D', MPS = 0).

G.2 Progressive decoding of the DCT

The description of the computation of the IDCT and the dequantization procedure contained in A.3.3 and A.3.4 apply to the progressive operation.

Progressive decoding processes must be able to decompress compressed image data which requires up to four sets of Huffman or arithmetic coder conditioning tables within a scan.

In order to avoid repetition, detailed flow diagrams of progressive decoder operation are not included. Decoder operation is defined by reversing the function of each step described in the encoder flow charts, and performing the steps in reverse order.

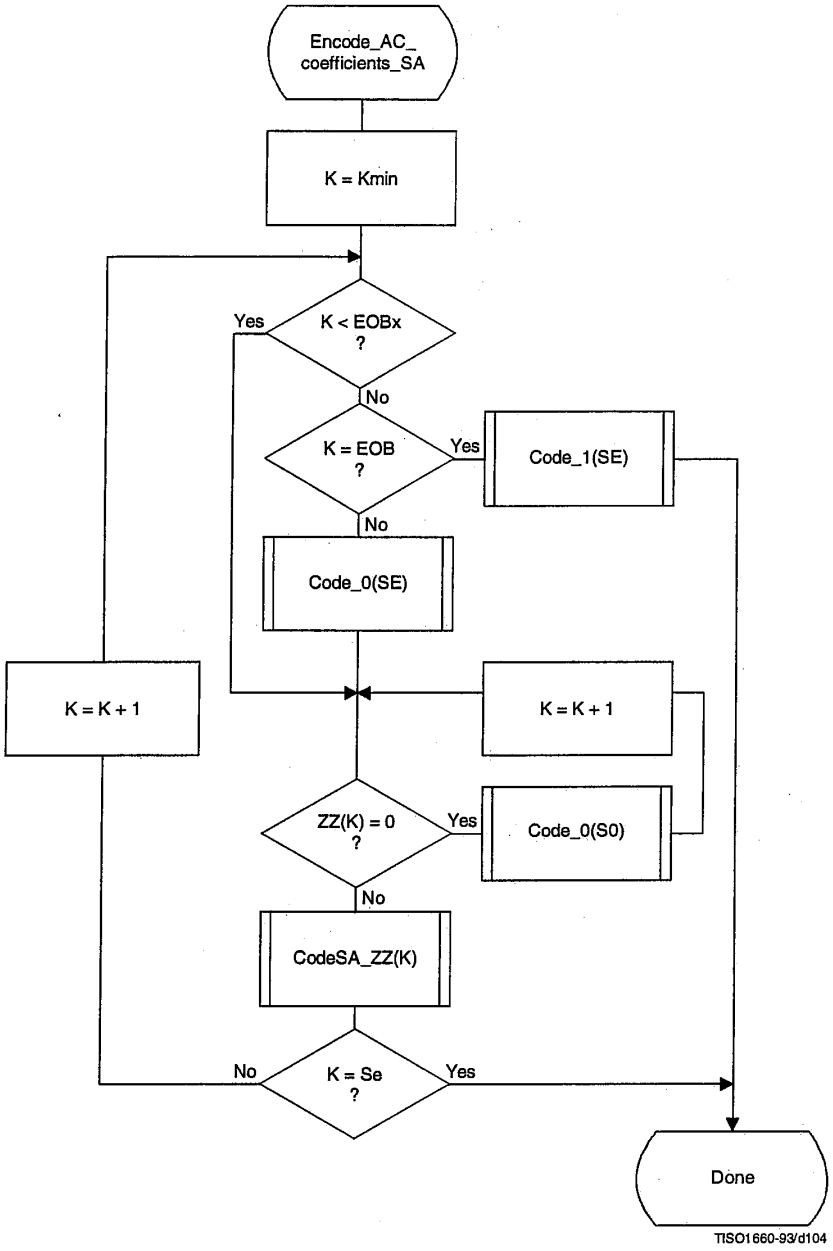


Figure G.10 – Subsequent successive approximation scans for coding of AC coefficients using arithmetic coding

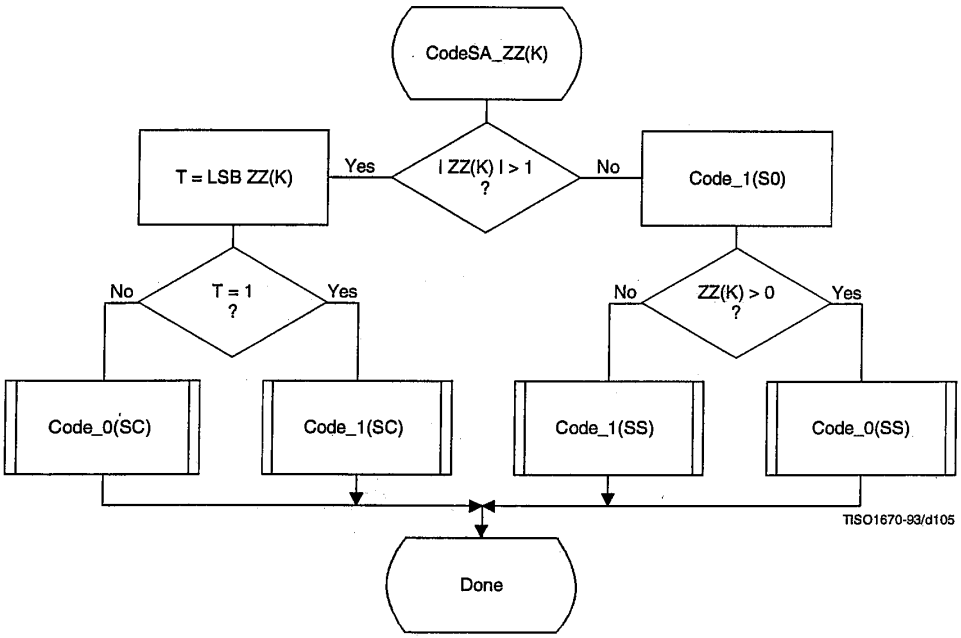


Figure G.11 – Coding non-zero coefficients for subsequent successive approximation scans

Table G.2 – Statistical model for subsequent scans of successive approximation coding of AC coefficient

Context-index	AC coding	Coding decision
SE	3 × (K–1)	K = EOB
S0	SE + 1	V = 0
SS	Fixed estimate	Sign
SC	S0 + 1	LSB ZZ(K) = 1

Annex H

Lossless mode of operation

(This annex forms an integral part of this Recommendation | International Standard)

This annex provides a **functional specification** of the following coding processes for the lossless mode of operation:

1) lossless processes with Huffman coding;

2) lossless processes with arithmetic coding.

For each of these, the encoding process is specified in H.1, and the decoding process is specified in H.2. The functional specification is presented by means of specific procedures which comprise these coding processes.

NOTE – There is **no requirement** in this Specification that any encoder or decoder which embodies one of the above-named processes shall implement the procedures in precisely the manner specified in this annex. It is necessary only that an encoder or decoder implement the **function** specified in this annex. The sole criterion for an encoder or decoder to be considered in compliance with this Specification is that it satisfy the requirements given in clause 6 (for encoders) or clause 7 (for decoders), as determined by the compliance tests specified in Part 2.

The processes which provide for sequential lossless encoding and decoding are not based on the DCT. The processes used are spatial processes based on the coding model developed for the DC coefficients of the DCT. However, the model is extended by incorporating a set of selectable one- and two-dimensional predictors, and for interleaved data the ordering of samples for the one-dimensional predictor can be different from that used in the DCT-based processes.

Either Huffman coding or arithmetic coding entropy coding may be employed for these lossless encoding and decoding processes. The Huffman code table structure is extended to allow up to 16-bit precision for the input data. The arithmetic coder statistical model is extended to a two-dimensional form.

H.1 Lossless encoder processes

H.1.1 Lossless encoder control procedures

Subclause E.1 contains the encoder control procedures. In applying these procedures to the lossless encoder, the data unit is one sample.

Input data precision may be from 2 to 16 bits/sample. If the input data path has different precision from the input data, the data shall be aligned with the least significant bits of the input data path. Input data is represented as unsigned integers and is not level shifted prior to coding.

When the encoder is reset in the restart interval control procedure (see E.1.4), the prediction is reset to a default value. If arithmetic coding is used, the statistics are also reset.

For the lossless processes the restart interval shall be an integer multiple of the number of MCU in an MCU-row.

H.1.2 Coding model for lossless encoding

The coding model developed for encoding the DC coefficients of the DCT is extended to allow a selection from a set of seven one-dimensional and two-dimensional predictors. The predictor is selected in the scan header (see Annex B). The same predictor is used for all components of the scan. Each component in the scan is modeled independently, using predictions derived from neighbouring samples of that component.

H.1.2.1 Prediction

Figure H.1 shows the relationship between the positions (a, b, c) of the reconstructed neighboring samples used for prediction and the position of x, the sample being coded.

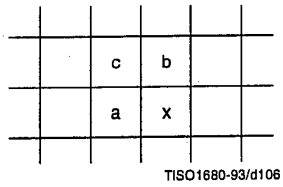


Figure H.1 – Relationship between sample and prediction samples

Define Px to be the prediction and Ra, Rb, and Rc to be the reconstructed samples immediately to the left, immediately above, and diagonally to the left of the current sample. The allowed predictors, one of which is selected in the scan header, are listed in Table H.1.

Table H.1 – Predictors for lossless coding

Selection-value	Prediction
0	No prediction (See Annex J)
1	Px = Ra
2	Px = Rb
3	Px = Rc
4	Px = Ra + Rb – Rc
5	Px = Ra + ((Rb – Rc)/2)[a)]
6	Px = Rb + ((Ra – Rc)/2)[a)]
7	Px = (Ra + Rb)/2

a) Shift right arithmetic operation

Selection-value 0 shall only be used for differential coding in the hierarchical mode of operation. Selections 1, 2 and 3 are one-dimensional predictors and selections 4, 5, 6, and 7 are two-dimensional predictors.

The one-dimensional horizontal predictor (prediction sample Ra) is used for the first line of samples at the start of the scan and at the beginning of each restart interval. The selected predictor is used for all other lines. The sample from the line above (prediction sample Rb) is used at the start of each line, except for the first line. At the beginning of the first line and at the beginning of each restart interval the prediction value of 2^{P-1} is used, where P is the input precision.

If the point transformation parameter (see A.4) is non-zero, the prediction value at the beginning of the first lines and the beginning of each restart interval is 2^{P-Pt-1}, where Pt is the value of the point transformation parameter.

Each prediction is calculated with full integer arithmetic precision, and without clamping of either underflow or overflow beyond the input precision bounds. For example, if Ra and Rb are both 16-bit integers, the sum is a 17-bit integer. After dividing the sum by 2 (predictor 7), the prediction is a 16-bit integer.

For simplicity of implementation, the divide by 2 in the prediction selections 5 and 6 of Table H.1 is done by an arithmetic-right-shift of the integer values.

The difference between the prediction value and the input is calculated modulo 2^{16}. In the decoder the difference is decoded and added, modulo 2^{16}, to the prediction.

H.1.2.2 Huffman coding of the modulo difference

The Huffman coding procedures defined in Annex F for coding the DC coefficients are used to code the modulo 2^{16} differences. The table for DC coding contained in Tables F.1 and F.6 is extended by one additional entry. No extra bits are appended after SSSS = 16 is encoded. See Table H.2.

Table H.2 – Difference categories for lossless Huffman coding

SSSS	Difference values
0	0
1	–1,1
2	–3,–2,2,3
3	–7..–4,4..7
4	–15..–8,8..15
5	–31..–16,16..31
6	–63..–32,32..63
7	–127..–64,64..127
8	–255..–128,128..255
9	–511..–256,256..511
10	–1 023..–512,512..1 023
11	–2 047..–1 024,1 024..2 047
12	–4 095..–2 048,2 048..4 095
13	–8 191..–4 096,4 096..8 191
14	–16 383..–8 192,8 192..16 383
15	–32 767..–16 384,16 384..32 767
16	32 768

H.1.2.3 Arithmetic coding of the modulo difference

The statistical model defined for the DC coefficient arithmetic coding model (see F.1.4.4.1) is generalized to a two-dimensional form in which differences coded for the sample to the left and for the line above are used for conditioning.

H.1.2.3.1 Two-dimensional statistical model

The binary decisions are conditioned on the differences coded for the neighbouring samples immediately above and immediately to the left from the same component. As in the coding of the DC coefficients, the differences are classified into 5 categories: zero(0), small positive (+S), small negative (–S), large positive (+L), and large negative (–L). The two independent difference categories combine to give 25 different conditioning states. Figure H.2 shows the two-dimensional array of conditioning indices. For each of the 25 conditioning states probability estimates for four binary decisions are kept.

At the beginning of the scan and each restart interval the conditioning derived from the line above is set to zero for the first line of each component. At the start of each line, the difference to the left is set to zero for the purposes of calculating the conditioning.

Difference to left (position a) \ Difference above (position b)	0	+S	−S	+L	−L
0	0	4	8	12	16
+S	20	24	28	32	36
−S	40	44	48	52	56
+L	60	64	68	72	76
−L	80	84	88	92	96

TISO1690-93/d107

Figure H.2 – 5 × 5 Conditioning array for two-dimensional statistical model

H.1.2.3.2 Assignment of statistical bins to the DC binary decision tree

Each statistics area for lossless coding consists of a contiguous set of 158 statistics bins. The first 100 bins consist of 25 sets of four bins selected by a context-index S0. The value of S0 is given by L_Context(Da,Db), which provides a value of 0, 4,..., 92 or 96, depending on the difference classifications of Da and Db (see H.1.2.3.1). The value for S0 provided by L_Context(Da,Db) is from the array in Figure H.2.

The remaining 58 bins consist of two sets of 29 bins, X1, ..., X15, M2, ..., M15, which are used to code magnitude category decisions and magnitude bits. The value of X1 is given by X1_Context(Db), which provides a value of 100 when Db is in the zero, small positive or small negative categories and a value of 129 when Db is in the large positive or large negative categories.

The assignment of statistical bins to the binary decision tree used for coding the difference is given in Table H.3.

Table H.3 – Statistical model for lossless coding

Context-index	Value	Coding decision
S0	L_Context(Da,Db)	V = 0
SS	S0 + 1	Sign
SP	S0 + 2	Sz < 1 if V > 0
SN	S0 + 3	Sz < 1 if V < 0
X1	X1_Context(Db)	Sz < 2
X2	X1 + 1	Sz < 4
X3	X1 + 2	Sz < 8
.	.	.
.	.	.
X15	X1 + 14	$Sz < 2^{15}$
M2	X2 + 14	Magnitude bits if Sz < 4
M3	X3 + 14	Magnitude bits if Sz < 8
.	.	.
.	.	.
M15	X15 + 14	Magnitude bits if $Sz < 2^{15}$

H.1.2.3.3 Default conditioning bounds

The bounds, L and U, for determining the conditioning category have the default values L = 0 and U = 1. Other bounds may be set using the DAC (Define-Arithmetic-Conditioning) marker segment, as described in Annex B.

H.1.2.3.4 Initial conditions for statistical model

At the start of a scan and at each restart, all statistics bins are re-initialized to the standard default value described in Annex D.

H.2 Lossless decoder processes

Lossless decoders may employ either Huffman decoding or arithmetic decoding. They shall be capable of using up to four tables in a scan. Lossless decoders shall be able to decode encoded image source data with any input precision from 2 to 16 bits per sample.

H.2.1 Lossless decoder control procedures

Subclause E.2 contains the decoder control procedures. In applying these procedures to the lossless decoder the data unit is one sample.

When the decoder is reset in the restart interval control procedure (see E.2.4) the prediction is reset to the same value used in the encoder (see H.1.2.1). If arithmetic coding is used, the statistics are also reset.

Restrictions on the restart interval are specified in H.1.1.

H.2.2 Coding model for lossless decoding

The predictor calculations defined in H.1.2 also apply to the lossless decoder processes.

The lossless decoders, decode the differences and add them, modulo 2^{16}, to the predictions to create the output. The lossless decoders shall be able to interpret the point transform parameter, and if non-zero, multiply the output of the lossless decoder by 2^{Pt}.

In order to avoid repetition, detailed flow charts of the lossless decoding procedures are omitted.

Annex J

Hierarchical mode of operation

(This annex forms an integral part of this Recommendation | International Standard)

This annex provides a **functional specification** of the coding processes for the hierarchical mode of operation.

In the hierarchical mode of operation each component is encoded or decoded in a non-differential frame. Such frames may be followed by a sequence of differential frames. A non-differential frame shall be encoded or decoded using the procedures defined in Annexes F, G and H. Differential frame procedures are defined in this annex.

The coding process for a hierarchical encoding containing DCT-based processes is defined as the highest numbered process listed in Table J.1 which is used to code any non-differential DCT-based or differential DCT-based frame in the compressed image data format. The coding process for a hierarchical encoding containing only lossless processes is defined to be the process used for the non-differential frames.

Table J.1 – Coding processes for hierarchical mode

Process	Non-differential frame specification	
1	Extended sequential DCT, Huffman, 8-bit	Annex F, process 2
2	Extended sequential DCT, arithmetic, 8-bit	Annex F, process 3
3	Extended sequential DCT, Huffman, 12-bit	Annex F, process 4
4	Extended sequential DCT, arithmetic, 12-bit	Annex F, process 5
5	Spectral selection only, Huffman, 8-bit	Annex G, process 1
6	Spectral selection only, arithmetic, 8-bit	Annex G, process 2
7	Full progression, Huffman, 8-bit	Annex G, process 3
8	Full progression, arithmetic, 8-bit	Annex G, process 4
9	Spectral selection only, Huffman, 12-bit	Annex G, process 5
10	Spectral selection only, arithmetic, 12-bit	Annex G, process 6
11	Full progression, Huffman, 12-bit	Annex G, process 7
12	Full progression, arithmetic, 12-bit	Annex G, process 8
13	Lossless, Huffman, 2 through 16 bits	Annex H, process 1
14	Lossless, arithmetic, 2 through 16 bits	Annex H, process 2

Hierarchical mode syntax requires a DHP marker segment that appears before the non-differential frame or frames. It may include EXP marker segments and differential frames which shall follow the initial non-differential frame. The frame structure in hierarchical mode is identical to the frame structure in non-hierarchical mode.

Either all non-differential frames within an image shall be coded with DCT-based processes, or all non-differential frames shall be coded with lossless processes. All frames within an image must use the same entropy coding procedure, either Huffman or arithmetic, with the exception that non-differential frames coded with the baseline process may occur in the same image with frames coded with arithmetic coding processes.

If the non-differential frames use DCT-based processes, all differential frames except the final frame for a component shall use DCT-based processes. The final differential frame for each component may use a differential lossless process.

If the non-differential frames use lossless processes, all differential frames shall use differential lossless processes.

For each of the processes listed in Table J.1, the encoding processes are specified in J.1, and decoding processes are specified in J.2.

NOTE – There is **no requirement** in this Specification that any encoder or decoder which embodies one of the above-named processes shall implement the procedures in precisely the manner specified by the flow charts in this annex. It is necessary only that an encoder or decoder implement the **function** specified in this annex. The sole criterion for an encoder or decoder to be considered in compliance with this Specification is that it satisfy the requirements given in clause 6 (for encoders) or clause 7 (for decoders), as determined by the compliance tests specified in Part 2.

In the hierarchical mode of operation each component is encoded or decoded in a non-differential frame followed by a sequence of differential frames. A non-differential frame shall use the procedures defined in Annexes F, G, and H. Differential frame procedures are defined in this annex.

J.1 Hierarchical encoding

J.1.1 Hierarchical control procedure for encoding an image

The control structure for encoding of an image using the hierarchical mode is given in Figure J.1.

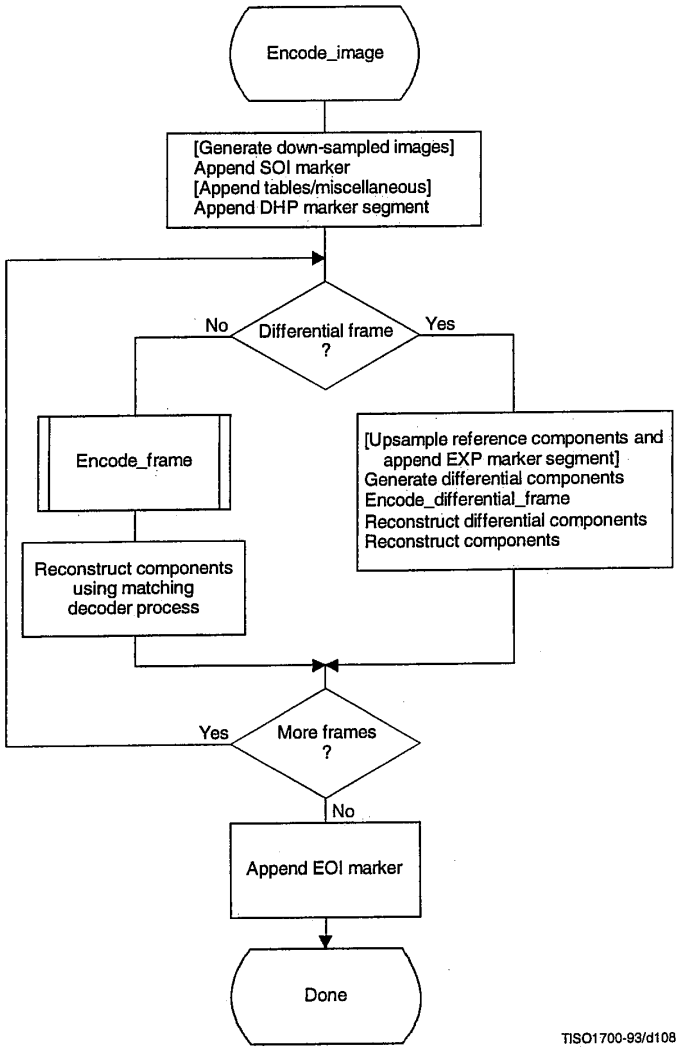


Figure J.1 – Hierarchical control procedure for encoding an image

In Figure J.1 procedures in brackets shall be performed whenever the particular hierarchical encoding sequence being followed requires them.

In the hierarchical mode the define-hierarchical-progression (DHP) marker segment shall be placed in the compressed image data before the first start-of-frame. The DHP segment is used to signal the size of the image components of the completed image. The syntax of the DHP segment is specified in Annex B.

The first frame for each component or group of components in a hierarchical process shall be encoded by a non-differential frame. Differential frames shall then be used to encode the two's complement differences between source input components (possibly downsampled) and the reference components (possibly upsampled). The reference components are reconstructed components created by previous frames in the hierarchical process. For either differential or non-differential frames, reconstructions of the components shall be generated if needed as reference components for a subsequent frame in the hierarchical process.

Resolution changes may occur between hierarchical frames in a hierarchical process. These changes occur if downsampling filters are used to reduce the spatial resolution of some or all of the components of the source image. When the resolution of a reference component does not match the resolution of the component input to a differential frame, an upsampling filter shall be used to increase the spatial resolution of the reference component. The EXP marker segment shall be added to the compressed image data before the start-of-frame whenever upsampling of a reference component is required. No more than one EXP marker segment shall precede a given frame.

Any of the marker segments allowed before a start-of-frame for the encoding process selected may be used before either non-differential or differential frames.

For 16-bit input precision (lossless encoder), the differential components which are input to a differential frame are calculated modulo 2^{16}. The reconstructed components calculated from the reconstructed differential components are also calculated modulo 2^{16}.

If a hierarchical encoding process uses a DCT encoding process for the first frame, all frames in the hierarchical process except for the final frame for each component shall use the DCT encoding processes defined in either Annex F or Annex G, or the modified DCT encoding processes defined in this annex. The final frame may use a modified lossless process defined in this annex.

If a hierarchical encoding process uses a lossless encoding process for the first frame, all frames in the hierarchical process shall use a lossless encoding process defined in Annex H, or a modified lossless process defined in this annex.

J.1.1.1 Downsampling filter

The downsampled components are generated using a downsampling filter that is not specified in this Specification. This filter should, however, be consistent with the upsampling filter. An example of a downsampling filter is provided in K.5.

J.1.1.2 Upsampling filter

The upsampling filter increases the spatial resolution by a factor of two horizontally, vertically, or both. Bi-linear interpolation is used for the upsampling filter, as illustrated in Figure J.2.

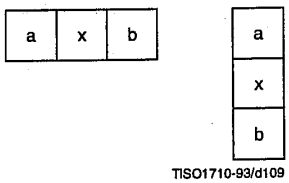


Figure J.2 – Diagram of sample positions for upsampling rules

The rule for calculating the interpolated value is:

$$P_x = (Ra + Rb) / 2$$

where Ra and Rb are sample values from adjacent positions a and b of the lower resolution image and Px is the interpolated value. The division indicates truncation, not rounding. The left-most column of the upsampled image matches the left-most column of the lower resolution image. The top line of the upsampled image matches the top line of the lower resolution image. The right column and the bottom line of the lower resolution image are replicated to provide the values required for the right column edge and bottom line interpolations. The upsampling process always doubles the line length or the number of lines.

If both horizontal and vertical expansions are signalled, they are done in sequence – first the horizontal expansion and then the vertical.

J.1.2 Control procedure for encoding a differential frame

The control procedures in Annex E for frames, scans, restart intervals, and MCU also apply to the encoding of differential frames, and the scans, restart intervals, and MCU from which the differential frame is constructed. The differential frames differ from the frames of Annexes F, G, and H only at the coding model level.

J.1.3 Encoder coding models for differential frames

The coding models defined in Annexes F, G, and H are modified to allow them to be used for coding of two's complement differences.

J.1.3.1 Modifications to encoder DCT encoding models for differential frames

Two modifications are made to the DCT coding models to allow them to be used in differential frames. First, the FDCT of the differential input is calculated without the level shift. Second, the DC coefficient of the DCT is coded directly – without prediction.

J.1.3.2 Modifications to lossless encoding models for differential frames

One modification is made to the lossless coding models. The difference is coded directly – without prediction. The prediction selection parameter in the scan header shall be set to zero. The point transform which may be applied to the differential inputs is defined in Annex A.

J.1.4 Modifications to the entropy encoders for differential frames

The coding of two's complement differences requires one extra bit of precision for the Huffman coding of AC coefficients. The extension to Tables F.1 and F.7 is given in Table J.2.

Table J.2 – Modifications to table of AC coefficient amplitude ranges

SSSS	AC coefficients
15	–32 767..–16 384, 16 384..32 767

The arithmetic coding models are already defined for the precision needed in differential frames.

J.2 Hierarchical decoding

J.2.1 Hierarchical control procedure for decoding an image

The control structure for decoding an image using the hierarchical mode is given in Figure J.3.

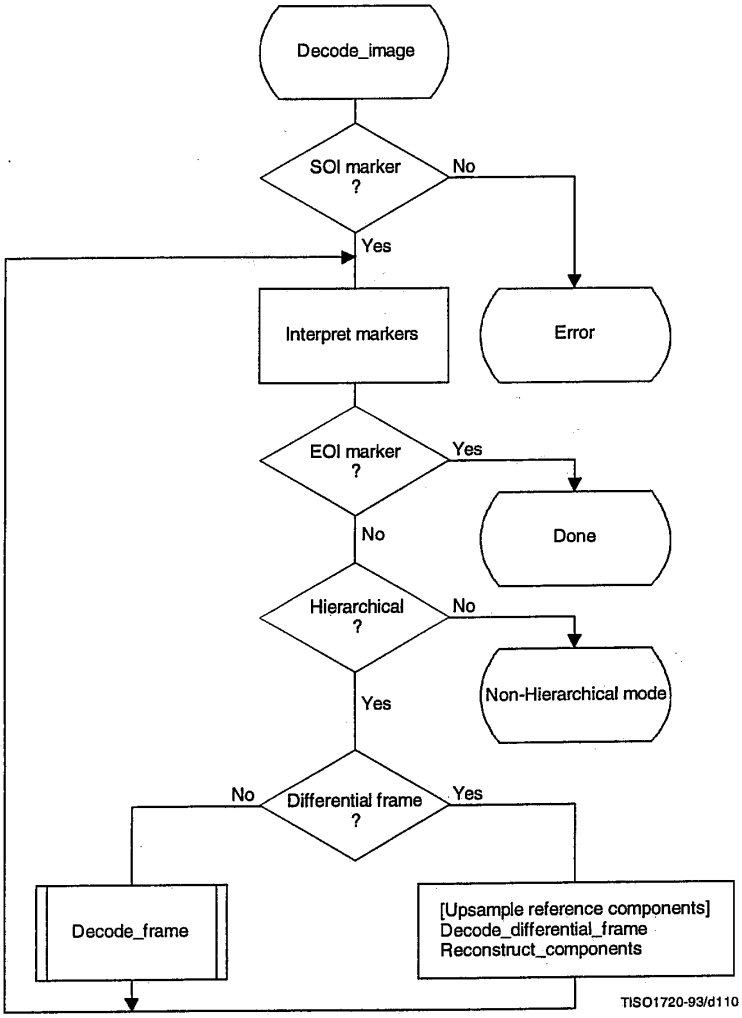


Figure J.3 – Hierarchical control procedure for decoding an image

The Interpret markers procedure shall decode the markers which may precede the SOF marker, continuing this decoding until either a SOF or EOI marker is found. If the DHP marker is encountered before the first frame, a flag is set which selects the hierarchical decoder at the "hierarchical?" decision point. In addition to the DHP marker (which shall precede any SOF) and the EXP marker (which shall precede any differential SOF requiring resolution changes in the reference components), any other markers which may precede a SOF shall be interpreted to the extent required for decoding of the compressed image data.

If a differential SOF marker is found, the differential frame path is followed. If the EXP was encountered in the Interpret markers procedure, the reference components for the frame shall be upsampled as required by the parameters in the EXP segment. The upsampling procedure described in J.1.1.2 shall be followed.

The Decode_differential_frame procedure generates a set of differential components. These differential components shall be added, modulo 2^{16}, to the upsampled reference components in the Reconstruct_components procedure. This creates a new set of reference components which shall be used when required in subsequent frames of the hierarchical process.

J.2.2 Control procedure for decoding a differential frame

The control procedures in Annex E for frames, scans, restart intervals, and MCU also apply to the decoding of differential frames and the scans, restart intervals, and MCU from which the differential frame is constructed. The differential frame differs from the frames of Annexes F, G, and H only at the decoder coding model level.

J.2.3 Decoder coding models for differential frames

The decoding models described in Annexes F, G, and H are modified to allow them to be used for decoding of two's complement differential components.

J.2.3.1 Modifications to the differential frame decoder DCT coding model

Two modifications are made to the decoder DCT coding models to allow them to code differential frames. First, the IDCT of the differential output is calculated without the level shift. Second, the DC coefficient of the DCT is decoded directly – without prediction.

J.2.3.2 Modifications to the differential frame decoder lossless coding model

One modification is made to the lossless decoder coding model. The difference is decoded directly – without prediction. If the point transformation parameter in the scan header is not zero, the point transform, defined in Annex A, shall be applied to the differential output.

J.2.4 Modifications to the entropy decoders for differential frames

The decoding of two's complement differences requires one extra bit of precision in the Huffman code table. This is described in J.1.4. The arithmetic coding models are already defined for the precision needed in differential frames.

Annex K

Examples and guidelines

(This annex does not form an integral part of this Recommendation | International Standard)

This annex provides examples of various tables, procedures, and other guidelines.

K.1 Quantization tables for luminance and chrominance components

Two examples of quantization tables are given in Tables K.1 and K.2. These are based on psychovisual thresholding and are derived empirically using luminance and chrominance and 2:1 horizontal subsampling. These tables are provided as examples only and are not necessarily suitable for any particular application. These quantization values have been used with good results on 8-bit per sample luminance and chrominance images of the format illustrated in Figure 13. Note that these quantization values are appropriate for the DCT normalization defined in A.3.3.

If these quantization values are divided by 2, the resulting reconstructed image is usually nearly indistinguishable from the source image.

Table K.1 – Luminance quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table K.2 – Chrominance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

K.2 A procedure for generating the lists which specify a Huffman code table

A Huffman table is generated from a collection of statistics in two steps. The first step is the generation of the list of lengths and values which are in accord with the rules for generating the Huffman code tables. The second step is the generation of the Huffman code table from the list of lengths and values.

The first step, the topic of this section, is needed only for custom Huffman table generation and is done only in the encoder. In this step the statistics are used to create a table associating each value to be coded with the size (in bits) of the corresponding Huffman code. This table is sorted by code size.

A procedure for creating a Huffman table for a set of up to 256 symbols is shown in Figure K.1. Three vectors are defined for this procedure:

FREQ(V)	Frequency of occurrence of symbol V
CODESIZE(V)	Code size of symbol V
OTHERS(V)	Index to next symbol in chain of all symbols in current branch of code tree

where V goes from 0 to 256.

Before starting the procedure, the values of FREQ are collected for V = 0 to 255 and the FREQ value for V = 256 is set to 1 to reserve one code point. FREQ values for unused symbols are defined to be zero. In addition, the entries in CODESIZE are all set to 0, and the indices in OTHERS are set to –1, the value which terminates a chain of indices. Reserving one code point guarantees that no code word can ever be all "1" bits.

The search for the entry with the least value of FREQ(V) selects the largest value of V with the least value of FREQ(V) greater than zero.

The procedure "Find V1 for least value of FREQ(V1) > 0" always selects the value with the largest value of V1 when more than one V1 with the same frequency occurs. The reserved code point is then guaranteed to be in the longest code word category.

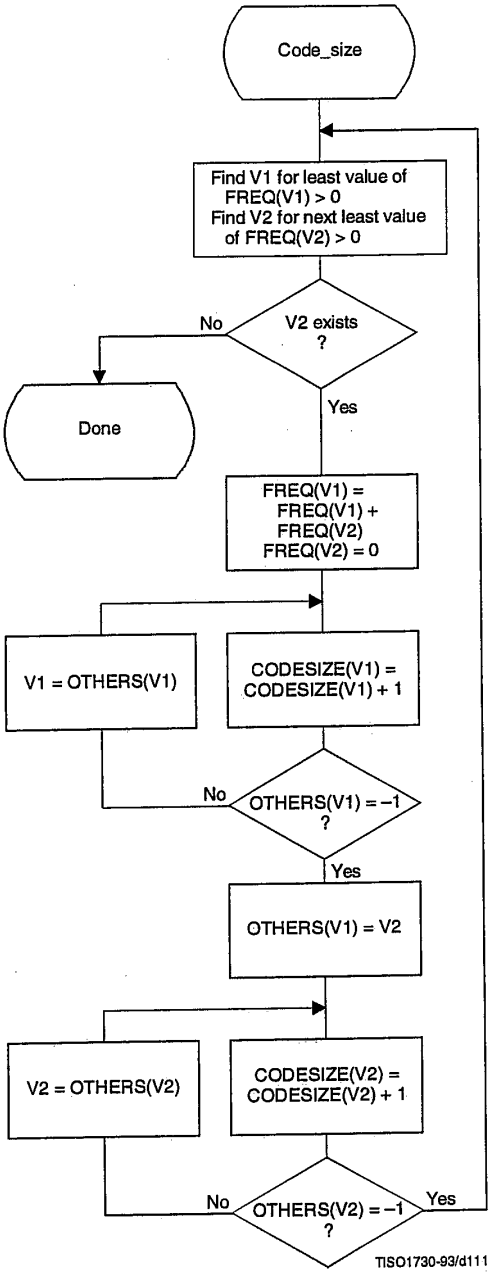


Figure K.1 – Procedure to find Huffman code sizes

Once the code lengths for each symbol have been obtained, the number of codes of each length is obtained using the procedure in Figure K.2. The count for each size is contained in the list, BITS. The counts in BITS are zero at the start of the procedure. The procedure assumes that the probabilities are large enough that code lengths greater than 32 bits never occur. Note that until the final Adjust_BITS procedure is complete, BITS may have more than the 16 entries required in the table specification (see Annex C).

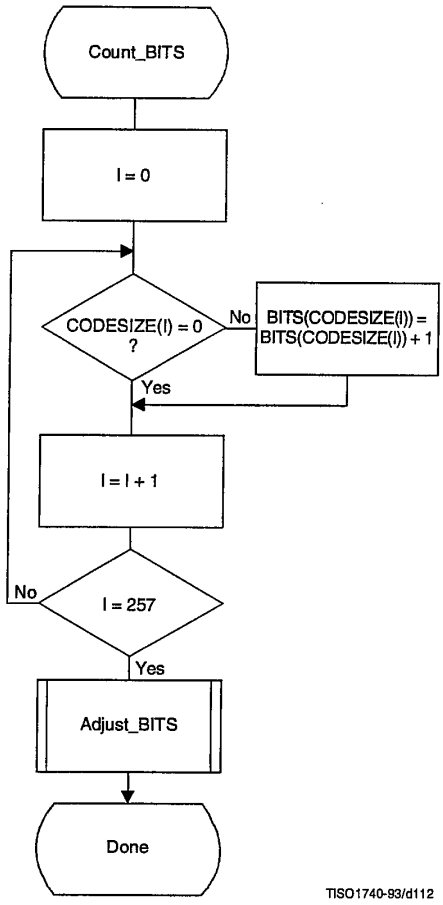


Figure K.2 – Procedure to find the number of codes of each size

Figure K.3 gives the procedure for adjusting the BITS list so that no code is longer than 16 bits. Since symbols are paired for the longest Huffman code, the symbols are removed from this length category two at a time. The prefix for the pair (which is one bit shorter) is allocated to one of the pair; then (skipping the BITS entry for that prefix length) a code word from the next shortest non-zero BITS entry is converted into a prefix for two code words one bit longer. After the BITS list is reduced to a maximum code length of 16 bits, the last step removes the reserved code point from the code length count.

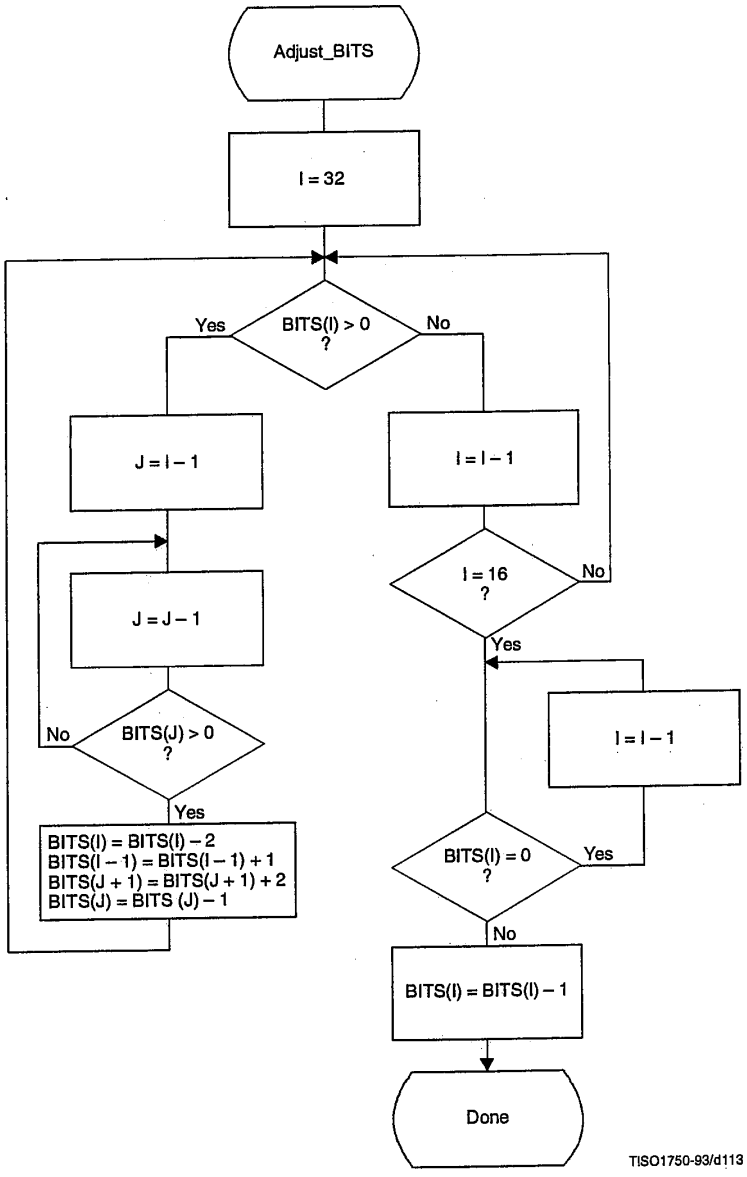


Figure K.3 – Procedure for limiting code lengths to 16 bits

The input values are sorted according to code size as shown in Figure K.4. HUFFVAL is the list containing the input values associated with each code word, in order of increasing code length.

At this point, the list of code lengths (BITS) and the list of values (HUFFVAL) can be used to generate the code tables. These procedures are described in Annex C.

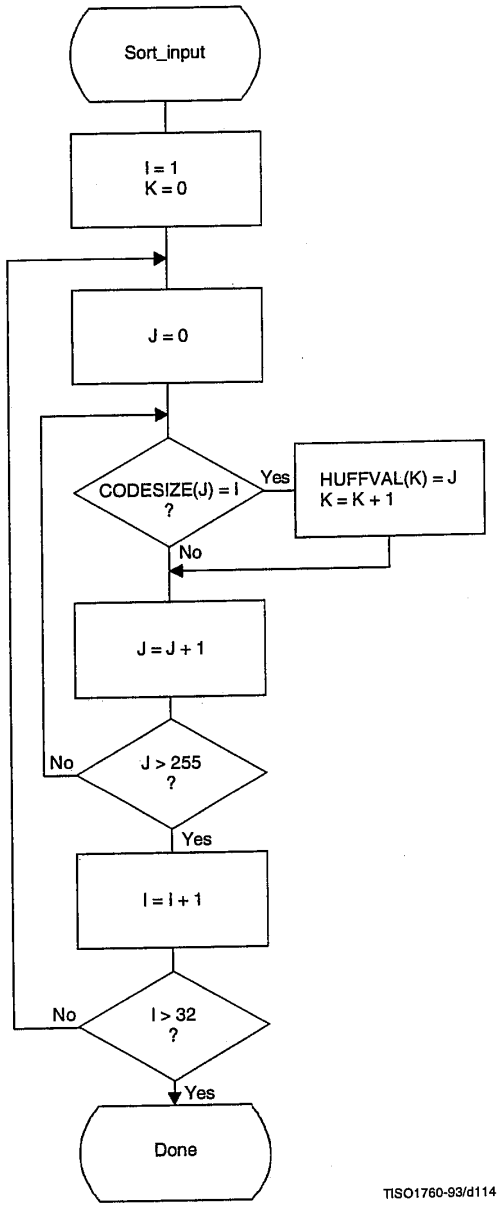


Figure K.4 – Sorting of input values according to code size

K.3 Typical Huffman tables for 8-bit precision luminance and chrominance

Huffman table-specification syntax is specified in B.2.4.2.

K.3.1 Typical Huffman tables for the DC coefficient differences

Tables K.3 and K.4 give Huffman tables for the DC coefficient differences which have been developed from the average statistics of a large set of video images with 8-bit precision. Table K.3 is appropriate for luminance components and Table K.4 is appropriate for chrominance components. Although there are no default tables, these tables may prove to be useful for many applications.

Table K.3 – Table for luminance DC coefficient differences

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table K.4 – Table for chrominance DC coefficient differences

Category	Code length	Code word
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

K.3.2 Typical Huffman tables for the AC coefficients

Tables K.5 and K.6 give Huffman tables for the AC coefficients which have been developed from the average statistics of a large set of images with 8-bit precision. Table K.5 is appropriate for luminance components and Table K.6 is appropriate for chrominance components. Although there are no default tables, these tables may prove to be useful for many applications.

Table K.5 – Table for luminance AC coefficients (sheet 1 of 4)

Run/Size	Code length	Code word
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	1111111110001001
2/6	16	1111111110001010
2/7	16	1111111110001011
2/8	16	1111111110001100
2/9	16	1111111110001101
2/A	16	1111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/4	16	1111111110001111
3/5	16	1111111110010000
3/6	16	1111111110010001
3/7	16	1111111110010010
3/8	16	1111111110010011
3/9	16	1111111110010100
3/A	16	1111111110010101

Table K.5 (sheet 2 of 4)

Run/Size	Code length	Code word
4/1	6	111011
4/2	10	1111111000
4/3	16	1111111110010110
4/4	16	1111111110010111
4/5	16	1111111110011000
4/6	16	1111111110011001
4/7	16	1111111110011010
4/8	16	1111111110011011
4/9	16	1111111110011100
4/A	16	1111111110011101
5/1	7	1111010
5/2	11	11111110111
5/3	16	1111111110011110
5/4	16	1111111110011111
5/5	16	1111111110100000
5/6	16	1111111110100001
5/7	16	1111111110100010
5/8	16	1111111110100011
5/9	16	1111111110100100
5/A	16	1111111110100101
6/1	7	1111011
6/2	12	111111110110
6/3	16	1111111110100110
6/4	16	1111111110100111
6/5	16	1111111110101000
6/6	16	1111111110101001
6/7	16	1111111110101010
6/8	16	1111111110101011
6/9	16	1111111110101100
6/A	16	1111111110101101
7/1	8	11111010
7/2	12	111111110111
7/3	16	1111111110101110
7/4	16	1111111110101111
7/5	16	1111111110110000
7/6	16	1111111110110001
7/7	16	1111111110110010
7/8	16	1111111110110011
7/9	16	1111111110110100
7/A	16	1111111110110101
8/1	9	111111000
8/2	15	111111111000000

Table K.5 (sheet 3 of 4)

Run/Size	Code length	Code word
8/3	16	1111111110110110
8/4	16	1111111110110111
8/5	16	1111111110111000
8/6	16	1111111110111001
8/7	16	1111111110111010
8/8	16	1111111110111011
8/9	16	1111111110111100
8/A	16	1111111110111101
9/1	9	111111001
9/2	16	1111111110111110
9/3	16	1111111110111111
9/4	16	1111111111000000
9/5	16	1111111111000001
9/6	16	1111111111000010
9/7	16	1111111111000011
9/8	16	1111111111000100
9/9	16	1111111111000101
9/A	16	1111111111000110
A/1	9	111111010
A/2	16	1111111111000111
A/3	16	1111111111001000
A/4	16	1111111111001001
A/5	16	1111111111001010
A/6	16	1111111111001011
A/7	16	1111111111001100
A/8	16	1111111111001101
A/9	16	1111111111001110
A/A	16	1111111111001111
B/1	10	1111111001
B/2	16	1111111111010000
B/3	16	1111111111010001
B/4	16	1111111111010010
B/5	16	1111111111010011
B/6	16	1111111111010100
B/7	16	1111111111010101
B/8	16	1111111111010110
B/9	16	1111111111010111
B/A	16	1111111111011000
C/1	10	1111111010
C/2	16	1111111111011001
C/3	16	1111111111011010
C/4	16	1111111111011011

Table K.5 (sheet 4 of 4)

Run/Size	Code length	Code word
C/5	16	1111111111011100
C/6	16	1111111111011101
C/7	16	1111111111011110
C/8	16	1111111111011111
C/9	16	1111111111100000
C/A	16	1111111111100001
D/1	11	11111111000
D/2	16	1111111111100010
D/3	16	1111111111100011
D/4	16	1111111111100100
D/5	16	1111111111100101
D/6	16	1111111111100110
D/7	16	1111111111100111
D/8	16	1111111111101000
D/9	16	1111111111101001
D/A	16	1111111111101010
E/1	16	1111111111101011
E/2	16	1111111111101100
E/3	16	1111111111101101
E/4	16	1111111111101110
E/5	16	1111111111101111
E/6	16	1111111111110000
E/7	16	1111111111110001
E/8	16	1111111111110010
E/9	16	1111111111110011
E/A	16	1111111111110100
F/0 (ZRL)	11	11111111001
F/1	16	1111111111110101
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

Removal of subjective redundancy from DCT-coded images

(7)

very - good
Mark again.

David L. McLaren, BE
D. Thong Nguyen, PhD

Indexing terms: Discrete cosine transform coding, Image processing, Subjective redundancy

Abstract: The removal of subjective redundancy from video images has recently become an important area of study. A suggested method of removing this redundancy from transform-coded images is through the psychovisual thresholding and quantisation of the image transform coefficients. In this paper, the coefficient thresholding and quantisation levels are based on the combined effects of spatial masking and the varying sensitivity of the human visual system to different spatial frequencies and levels of luminance. By combining the Discrete Cosine Transform (DCT) method of image coding with psychovisual thresholding and quantisation schemes, subdistortion motion video bit-rates as low as 2.5 Mbit/s (non-interlaced 25 frame-per-second video) have been obtained without the need for interframe coding.

1 Introduction

The increasing user demand for video as a communication medium over the last decade has greatly increased the need for efficient image coding and compression methods. Although many data compression algorithms have been proposed in the past, only recently have high-compression algorithms been introduced. The first coding schemes, involving simple Differential Pulse Code Modulation (DPCM) and Adaptive Predictive Coding (APC) algorithms, were only able to obtain compression ratios of up to 2.5 : 1 [1]. Interpolative and extrapolative coding went a step further and increased the compression ratio to around 4 : 1 [1] by transmitting only a subset of the samples and interpolating or extrapolating to obtain the full image. However, the most recent and most successful methods of image compression to date have been transform-coding-based [2, 3]. By transforming spatial data into another domain (usually frequency-related), statistical independence between pixels and high-energy compaction can be obtained. In particular, the Discrete Cosine Transform (DCT) algorithm has become widely recognised as an almost optimum transform method when compared with other transforms on the basis of energy compaction and decorrelation between pixels [4, 5].

The general method of discrete cosine transform coding [5] involves dividing the original spatial image into smaller $N \times N$ blocks of pels, and then transforming the blocks to obtain equal-sized blocks of transform coefficients in the frequency domain. These coefficients are then thresholded, quantised and coded ready for transmission. By combining the discrete cosine transform with a minimum redundancy coding scheme [5] much of the statistical redundancy in an image can be removed in the coding process. Recently, however, the removal of subjective redundancy, through the thresholding and quantising of the transform coefficients, has also become an important area of study as the quest continues to further reduce the required bit rates to transmit still and moving images. The problem has, however, not been dealt with adequately.

It is the removal of these subjective redundancies from DCT-coded images, through psychovisual thresholding and quantisation, which is the subject of this paper.

The compression techniques described in this paper are all general in nature and are therefore applicable to the coding and compression of any image or video-based service from low bit-rate video-telephony to High Definition Television (HDTV). The sub-distortion results presented in Section 5 are, however, more suitable for intermediary services such as high-quality video conferencing or low-quality entertainment television with bit-rates in the region of 1 to 5 Mbit/s.

2 Subjective redundancy

Unlike statistical redundancy, the removal of subjective redundancy is an irreversible process and involves discarding information which the designer feels can be removed without any change being noticed by the human observer [6]. The sensitivity of the human visual system to stimuli of varying levels of contrast, luminance and different spatial and temporal frequencies varies greatly [6], and these inconsistencies can be exploited to determine how information can be discarded without subjectively degrading the final image. A number of methods have already been proposed for including certain psychovisual properties of the human visual system (frequency sensitivity [7, 8], luminance dependence [6] and masking effects [9, 6]) into image coding and compression schemes. However, no coding scheme has yet adequately combined these effects to produce a simple, efficient and optimum method of removing subjectively-redundant information.

There are two areas in the standard transform-coding process — the thresholding and the quantising of the DCT coefficients — where the subjective redundancy in an image, and hence the number of bits required for representation, can be reduced.

Many of the DCT coefficients, obtained by transforming the blocks of spatial image, are small enough not

Paper 8175I (E4), first received 20th December 1989 and in revised form 21st March 1991

The authors are with the Department of Electrical and Electronic Engineering, University of Tasmania, GPO Box 252C, Hobart, Tasmania 7001, Australia

to be transmitted. By thresholding the blocks of coefficients, values below a given threshold level will be set to zero leaving a reduced number of coefficients for coding. Of course, as more coefficients are set to zero the quality of the reconstructed image deteriorates. However, the way in which the image quality is affected depends not only on the number of non-zero coefficients retained but also on which coefficients are discarded. Harsh thresholding of low-frequency coefficients causes blocking effects (sub-block boundaries becoming visible), while dropping too many high-frequency coefficients results in a loss of resolution and blurring in areas of high activity. For this reason, an $N \times N$ thresholding matrix is used which is made up of optimum thresholding values for each spatial frequency, and which removes only subjectively-redundant coefficients.

Once the blocks of coefficients have been thresholded, the remaining non-zero coefficients are quantised to reduce the number of levels and hence further reduce the number of bits. Again, over-harsh quantisation of coefficients corresponding to different spatial frequencies affects the reconstructed image in different ways. Over-quantising low-frequency coefficients again causes blocking, while large quantisation steps at higher frequencies lead to random noise becoming visible. The phenomenon of spatial masking can also be taken into account to allow for larger quantisation step sizes in certain areas.

In the past, these coefficient thresholding and quantising stages have been combined into a single uniform quantisation scheme where only those coefficients below the lowest quantisation step size are discarded [5]. However, because harshly thresholding and quantising different transform coefficients leads to different subjective effects, these two areas should be treated separately.

3 Psychovisual thresholding

Because of the varied effects of harshly thresholding DCT coefficients of different spatial frequencies, it is clear that a constant threshold level for all coefficients is not efficient. When a typical video image (3 : 4 aspect ratio) is viewed from a standard viewing distance [6], the spatial frequency, w_{ij}, in cycles per degree (cpd), of a coefficient, c_{ij}, can be calculated from

$$w_{ij} = \sqrt{\left[\left(\frac{32i}{N-1}\right)^2 + \left(\frac{24j}{N-1}\right)^2\right]}$$

$$i, j = 0, 1, 2, \ldots, N-1$$

where N is the sub-block size, and i and j are the matrix row and column indices respectively. Psychovisual studies have shown that the human visual system has a general bandpass characteristic [10, 7] with peak sensitivity between 3 and 4 cycles per degree and reduced sensitivity at higher and lower spatial frequencies (Fig. 1). This response curve has been the subject of much research in the past and, as a result, a fairly standard transfer function has evolved. One of the more common forms of this sensitivity function, S_{ij}, proposed by Ngan in [9], is given in eqn. 1

$$S_{ij} = (0.31 + 0.69 w_{ij}) e^{-0.29 w_{ij}}$$

$$i, j = 0, 1, 2, \ldots, N-1 \quad (1)$$

By making the coefficient thresholding levels inversely proportional to the relative sensitivities of the corresponding spatial frequencies, coefficients corresponding to relatively insensitive frequencies will be more harshly thresholded than those corresponding to frequencies of higher sensitivity. However, the spatial frequency sensitivity function of eqn. 1 has been constructed from subjective tests where the distribution of energy is uniform over all frequencies [7]. As this is not true for the blocks of DCT coefficients, the sensitivity curve must be normalised by the average power at each frequency.

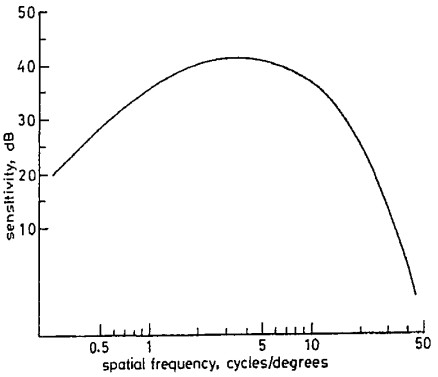


Fig. 1 *Frequency sensitivity curve*

To determine the coefficient energy distribution, the power in each coefficient was averaged over each sub-block in ten different 512 × 512 × 8-bit images (10240 blocks in all) to obtain the distribution in Fig. 2. This

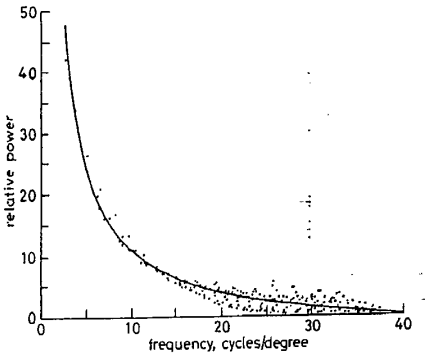


Fig. 2 *Coefficient power distribution*

energy distribution can be adequately modelled by the 'best-fit' function of eqn. 2 (shown by the solid line in Fig. 2)

$$P_{ij} = 34.10 \times w_{ij}^{-0.94} - 1 \quad i, j = 0, 1, 2, \ldots, N-1 \quad (2)$$

In addition to varying with spatial frequency content, the sensitivity of the human visual system to small changes in a single sub-block is directly proportional to the average background luminance of the block. This relationship is known as Weber's Law [11] and, although it is slightly distorted by the non-linear relationship between the applied voltage and the displayed luminance of a typical television screen, it still holds at high luminance levels [6]. As the DC transform coefficient, c_{00}, is a measure of the average luminance in an image [5], this effect is easily incorporated into the coding process by simply scaling each block thresholding matrix by c_{00}.

The $N \times N$ matrix of sensitivity values, S_{ij}, is normalised using the power distribution, P_{ij}, and each value is

inverted to obtain a normalised sensitivity matrix S'_{ij}, $i, j = 0, 1, 2, \ldots, N-1$. This matrix is given in Table 1 for a sub-block size of 16 × 16 pels. Each value in the matrix is then multiplied by c_{00}, and uniformly scaled so that

Table 1: Thresholding level matrix, S'_{ij}

0	2	2	1	1	1	1	1	1	2	2	3	4	4	6
2	2	1	1	1	1	1	1	1	2	2	3	4	4	6
1	1	1	1	1	1	1	1	1	2	2	3	4	5	6
1	1	1	1	1	1	1	1	1	2	2	3	3	4	5	7
1	1	1	1	1	1	1	1	2	2	2	3	4	4	6	7
1	1	1	1	1	1	1	2	2	2	3	4	4	5	7	8
1	1	1	1	1	2	2	2	2	3	3	4	5	6	8	9
2	2	2	2	2	2	2	3	3	4	4	5	6	8	9	11
2	2	2	2	2	3	3	3	4	4	5	6	8	9	11	13
3	3	3	3	3	4	4	4	5	6	7	8	9	11	13	17
4	4	4	4	4	5	5	6	7	8	9	11	13	15	17	19
5	5	6	6	6	7	7	8	9	9	11	13	15	17	19	25
8	8	8	8	8	9	9	11	13	13	15	17	19	22	25	27
11	11	11	11	11	13	13	15	17	17	19	22	25	27	30	32
15	15	15	15	17	17	17	19	19	22	25	27	27	30	32	33
19	19	19	19	22	22	22	25	25	27	30	30	32	33	32	28

the thresholding, although as harsh as possible, still causes only sub-threshold distortion (unseen by the human observer). All subjective scaling and testing is performed as a recursive comparison procedure [12], where the parameter in question is adjusted until no visible difference can be seen between the original and reconstructed images when viewed from a standard viewing distance of 6 to 8 times the image height [6]. Several independent subjects were also used in each of these viewing sessions. To avoid blocking effects, the low-frequency coefficients (below 5 cycles per degree) are further reduced to suitable values, T_0 (again determined through subjective tests as described above). This final matrix of thresholding values, T_{ij}, (given by eqn. 3), is then used to threshold the blocks of DCT coefficients before quantisation.

$$T_{ij} = \begin{cases} A\left(\dfrac{S_{ij}}{P_{ij}}\right)^{-1} c_{00} & w_{ij} \geq 5 \text{ cpd} \\ T_0 & w_{ij} < 5 \text{ cpd} \end{cases} \qquad i, j = 0, 1, 2, \ldots, N-1 \quad (3)$$

It is important to note that although each image sub-block is thresholded by the same basic matrix (T_{ij}), the varying amount of activity in each block (reflected in the relative magnitudes of the DCT coefficients), combined with the changing luminance values (c_{00}), makes this thresholding scheme inherently adaptive to changing image characteristics. Sub-blocks containing little or no information (and hence very small non-DC transform coefficients) are thresholded relatively more harshly and produce fewer bits for transmission.

4 Psychovisual quantisation

Once thresholded, the remaining coefficients are quantised to a number of discrete levels. To make the thresholding and quantisation levels independent, the lowest quantisation level, $q_{ij}^{(1)}$, is set half a step above the threshold level, T_{ij}. The overall quantisation scheme is then uniform from that point, as shown in Fig. 3. The nonzero transform coefficients, c_{ij}, are then quantised to $\tilde{c}_{ij}$ using

$$\tilde{c}_{ij} = \begin{cases} \left\lfloor \dfrac{c_{ij} - T_{ij} + Q_{ij}/2}{Q_{ij}} \right\rfloor & c_{ij} > 0 \\ \left\lfloor \dfrac{c_{ij} + T_{ij} - Q_{ij}/2}{Q_{ij}} \right\rfloor & c_{ij} < 0 \end{cases} \qquad i, j = 0, 1, 2, \ldots, N-1$$

where $\lfloor x \rfloor$ refers to the integer value closest to x. The optimum quantisation step sizes for each coefficient, Q_{ij}, again depend on the spatial frequency sensitivity curve.

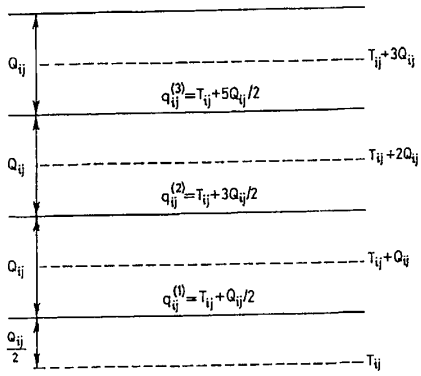


Fig. 3 *Coefficient quantisation scheme*

The effects of spatial masking, however, can also be exploited to allow for larger quantisation step sizes in DCT blocks containing areas of high activity [13].

Spatial masking is a well known phenomenon [6, 14] and refers to the changing visibility of a single stimulus in an area of varying spatial and temporal activity. In a still image, this leads to a reduction in the visibility of pixel errors in areas of high-detail luminance changes (high activity). The relationship between the allowable quantisation step size for sub-threshold distortion and the amount of activity in a block has been the subject of previous research [6]. For uniform quantisation, the relationship is given by

$$Q_{ij} = \sqrt{(T_{ij} A_F)} \quad i, j = 0, 1, 2, \ldots, N-1 \qquad (4)$$

where T_{ij} is the threshold level corresponding to the spatial frequency at matrix co-ordinates (i, j) and A_F is the block activity function* (a measure of the amount of activity in a block).

In References 6 and 14 A_F is defined as the sum of first derivatives in the spatial domain. However, this definition produces a number of inconsistencies. For example, a ramp and a sawtooth function would give the same value for A_F. In this paper, we propose a more accurate measure of block activity i.e. the power contained in the sum of second derivatives in the spatial domain. The Laplacian edge detector [15] achieves second-order differentiation through the approximation

$$w^2 = (w_1^2 + w_2^2) \approx 4 - 2\cos w_1 - 2\cos w_2$$

Converting to the (z_1, z_2)-domain, this approximation results in the well-known Laplacian mask, L, in eqn. 5. To take into account the 3 : 4 aspect ratio of most video images, this mask is altered to obtain the mask, M, in eqn. 6.

$$L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad (5)$$

$$M = \begin{bmatrix} -1 & -2.777 & -1 \\ -1.5625 & 12.679 & -1.5625 \\ -1 & -2.777 & -1 \end{bmatrix} \qquad (6)$$

* Also known as the masking function.

The activity function, $A_F > 1$, is then given by

$$A_F = 1 + q\sqrt{\left(\sum_{i,j=0}^{N-1} (M * * X)_{ij}^2\right)} \quad (7)$$

where $M * * X$ denotes the 2-dimensional convolution output of the edge operator, q is a normalisation factor and N is the sub-block dimension. The square-root of the summation has been applied to express the power in linear units. However, when employing transform coding, an ideal activity function should be calculated directly from the blocks of transform coefficients. By invoking Parseval's theorem, the total power contained in the summation of second derivatives in the spatial domain can be transformed into the frequency domain to obtain

$$\sum_{all\ m} \left|\frac{d^2x(m)}{dm^2}\right|^2 = \sum_{i,j=0}^{N-1} (w_{ij}^2 c_{ij})^2$$

Taking the square-root of the summation, again to revert to linear units, an activity function in the transform domain is given by

$$A_F = 1 + q\sqrt{\left(\sum_{i,j=0}^{N-1} w_{ij}^4 c_{ij}^2\right)} \quad (8)$$

where w_{ij} is the spatial frequency corresponding to matrix position (i, j). The activity functions given by eqns. 7 and 8 are, however, computationally costly. In view of the asymmetry of the mask M, A_F as given in eqn. 7 requires $(4N^2 + 1)$ multiplications per block while, by using a pre-calculated w_{ij}^4 matrix, A_F as defined in eqn. 8 requires $(2N^2 + 1)$ multiplications per block. If a simplifying approximation to eqn. 8 is made, which includes the square-root in the summation, the equation reduces to

$$A_F = 1 + q\sum_{i,j=0}^{N-1} w_{ij}^2 |c_{ij}| \quad (9)$$

Eqn. 9 now produces results which are well correlated with those produced by eqn. 7 (shown by the correlation plot of Fig. 4) and at a much reduced computational cost (now only $(N^2 + 1)$ multiplications per block and no square-root operator). Eqn. 9, therefore, provides an alternative definition for A_F which can be used in situations where the advantages of increased computational efficiency outweigh the disadvantages of reduced accuracy.

By combining the subjective thresholding matrix, T_{ij}, with the activity function, as described by eqn. 4, the coefficient quantisation step sizes are obtained. The normalisation factor, q, is again subjectively adjusted (using the subjective testing criteria described in Section 3) so that quantisation of the blocks of coefficients results in only sub-threshold distortion.

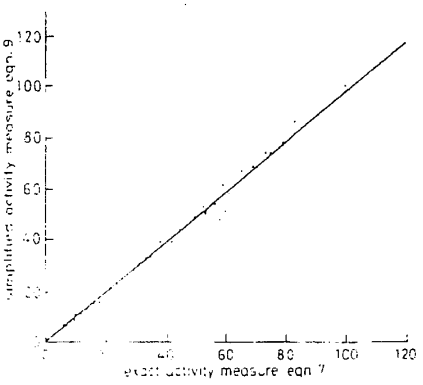


Fig. 4 *Activity measure correlation plot*

5 Results

The psychovisual thresholding and quantisation schemes, described in Sections 3 and 4 respectively, have been combined with the standard DCT-coding algorithm and applied to two 512 × 512 × 8-bit images (Figs. 5*a* and

Fig. 5 *'Face' image*
a Original image
b Reconstructed image
c Scaled difference image

6a). A sub-block size of 16 × 16 pels was used ($N = 16$) and the images were thresholded and quantised as harshly as possible while avoiding supra-threshold distortion (no visible differences between the original and reconstructed images when viewed from a standard viewing distance [6]). The resulting blocks of transform coefficients were scanned in order of increasing frequency, and Huffman-coded using tables of optimum code-words [5]. The compression results obtained (bits/pel ratios and bit-rates for non-interlaced 25 frame-per-second video) are summarised in Table 2 along with compression results obtained without the use of subjective compression techniques (transform coding without any thresholding or quantising of the transform coefficients). Both the reconstructed and difference (between original and reconstructed) images and are displayed in Figs. 5b and 6b and Figs. 5c and 6c, respectively. The difference images have been scaled by a factor of five to make the variations visible. A darker area indicates no change.

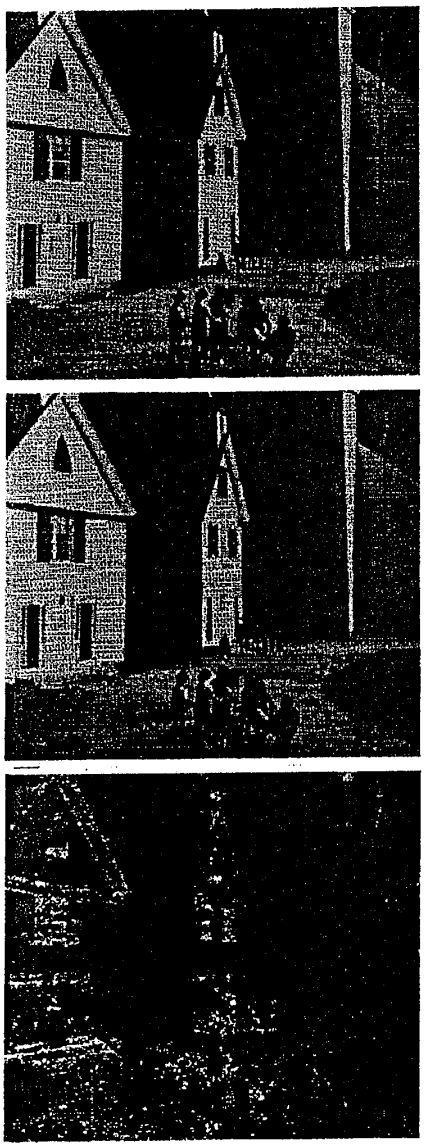


Fig. 6 *'Church' image*
a Original image
b Reconstructed image
c Scaled difference image

Table 2: Compression results

Image	Standard DCT compression		Perceptually optimum compression	
	Compression	Bit-rate	Compression	Bit-rate
Face	1.47 bit/pel	9.61 Mbit/s	0.38 bit/pel	2.51 Mbit/s
Church	2.39 bit/pel	15.65 Mbit/s	0.66 bit/pel	4.30 Mbit/s

As expected, the higher complexity 'church' image requires a higher bit-rate for transmission than the simpler head-and-shoulders image. By removing most of the subjective redundancy from the two test images, compression ratios up to 0.38 bits per pel (21 : 1) have been obtained without the need for interframe coding. This is also an improvement by a factor of 3.8 when compared to the standard DCT-coding algorithm without psychovisual compression.

6 Conclusions

The combination of standard transform coding techniques and psychovisually optimum thresholding and quantising schemes has resulted in an optimum, high-compression, image-coding algorithm. Because of the general nature of the psychovisual effects exploited in the compression scheme, the same techniques can be incorporated into almost any image communication system involving still or moving images.

Although the results in Table 2 are optimum for subthreshold distortion, it should be remembered that the bit-rates could be further reduced if a limited amount of suprathreshold distortion was allowed for a lower grade of service during periods of network congestion. Also, the thresholding and quantisation levels in this case have been optimised for still images. The sensitivity of the human visual system to different spatial frequencies is greatly reduced over the entire spectrum as the temporal frequency approaches that of motion pictures [6], in which case the images could be thresholded and quantised more harshly. The result would be even lower bit-rates while still retaining a high image quality.

7 References

1 KUNT, M., IKONOMOPOULOS, A., and KOCHER, M.: 'Second-generation image-coding techniques', *Proc. IEEE*, 1985, **73**, (4), pp. 540–574
2 NGUYEN, D.T., CHUA, K.C., and McLAREN, D.L.: 'Bandwidth allocation for packet video signals in ATM multiplexers'. Fourth ATERB Fast Packet Switching Workshop Proceedings, Sydney, July 1989
3 JAIN, A.K.: 'Image data compression: a review', *Proc. IEEE, 1981*, **69**, (3), pp. 349–389
4 PERKINS, M.G.: 'A comparison of the Hartley, Cas-Cas, Fourier, and discrete cosine transforms for image coding', *IEEE Trans.*, 1988, **COM-36**, (6), pp. 758–761
5 CHEN, W.H., and PRATT, W.K.: 'Scene adaptive coder', *IEEE Trans.* 1984, **COM-32**, (3), pp. 225–232
6 NETRAVALI, A.N., and HASKELL, B.G.: 'Digital pictures: representation and compression' (Plenum Press, New York, 1988)
7 SAKRISON, D.J.: 'On the role of the observer and a distortion measure in image transmission', *IEEE Trans.*, 1977, **COM-25**, (11), pp. 1251–1267

8 CHITPRASERT, B., and RAO, K.R.: 'Human visual weighted progressive image transmission', *IEEE Trans.*, 1990, COM-38, (7), pp. 1040–1044
9 NGAN, K.N., LEONG, K.S., and SINGH, H.: 'Adaptive cosine transform coding of images in perceptual domain', *IEEE Trans.*, 1989, ASSP-37, (11), pp. 553–559
10 WILSON, H.R.: 'Quantitative prediction of line spread function measurements: implications for channel bandwidths', *Vis. Res.*, 1977, 18, (4), pp. 493–496
11 CORNSWEET, T.N.: 'Visual perception' (Academic Press, New York, 1971)
12 PEARSON, D.E.: 'Transmission and display of pictorial information' (Pentech Press, London, 1975)
13 McLAREN, D.L., and NGUYEN, D.T.: 'Activity function for DCT coded images', *Electron. Lett.*, 1989, 25, (25), pp. 1704–1705
14 NETRAVALI, A.N., and PRASADA, B.: 'Adaptive quantization of picture signals using spatial masking', *Proc. IEEE*, 1977, 65, (4), pp. 536–548
15 NGUYEN, D.T.: 'A unified approach to differential edge detectors'. 23rd New Zealand National Electronic Conf. Proc., August 1986, Palmerston North, pp. 161–167

42.2: Color-Facsimile System for Mixed-Color Documents

I. Miyagawa, H. Mizumachi, M. Matsuki
NTT Human Interface Laboratories, Kanagawa, Japan

ABSTRACT

In order to transmit mixed color documents such as a color page of illustrated magazine, with high data efficiency, we proposed to use ODA type document structure and developed the simulation system. The result of coding simulation, automatic image area separation and the simulation system are described.

INTRODUCTION

Due to recent progress in color image technology, the demand for color image communication such as color facsimile is rapidly increasing. Color documents to be transmitted by color facsimile can be roughly classified into the following types:

- multi color : color pie graphs, B&W(Black and White) documents marked with red ink,
- full color : color photographs,
- mixed color : combinations of above documents (ex. color page of illustrated magazine, color catalog)

Standardization of a color extension for facsimile is being discussed in ITU-T Study Group 8. The representation method for a single full color image on a single page will be developed as the first step; mixed colors will be the second step. As many color documents can be classified as mixed color documents, a highly efficient but high quality encoding method for mixed color documents is very important.

We proposed to use an ODA(Open Document Architecture)[3] type document structure such as page and block, and single content type i.e. raster graphic content, for the encoding of mixed color documents[1,2]. We have developed a simulation system for this encoding and communication system named the Mixed Color Facsimile. This paper presents results gained from encoding simulations, mixed color syntax, and automatic image area separation, especially for photographic images and B&W documents.

MIXED COLOR FACSIMILE

In the past few years, remarkable advances have been made in the development and standardization of image coding techniques. The JPEG[4] and JBIG[5] encoding schemes were developed by an ITU-T and ISO/IEC joint group. The JPEG encoding scheme was developed for full color images. The JBIG encoding scheme was developed for B&W bi-level images and bit-plane images such as multi-color images. Therefore, they are not suitable for other types of images.

If only one encoding scheme, JPEG or JBIG, is used for mixed color documents, we may not be able to achieve high efficiency and high quality for all document components.

In order to solve this issue, we introduce the mixed color communication mode, in which a page of mixed color components is divided into few different image types such as full color, multi-color, and B&W binary. Each type is encoded using the most suitable encoding method. Full color areas are encoded using JPEG. Multi-color areas and B&W binary images are encoded by JBIG or MMR. For example, if the test image containing a full color component (JPEG gold hill) and a B&W document (CCITT Test document NO.4) in one page as shown in Figure 1 is encoded by JPEG, the result is a compressed file of 500kbytes. If this image is divided into a full-color area and a B&W bi-level area and coded with JPEG and MMR or JBIG respectively, the compressed file occupies only 125 to 129 kbytes. This is about one fourth that output by JPEG only. The result is summarized in Table 1.

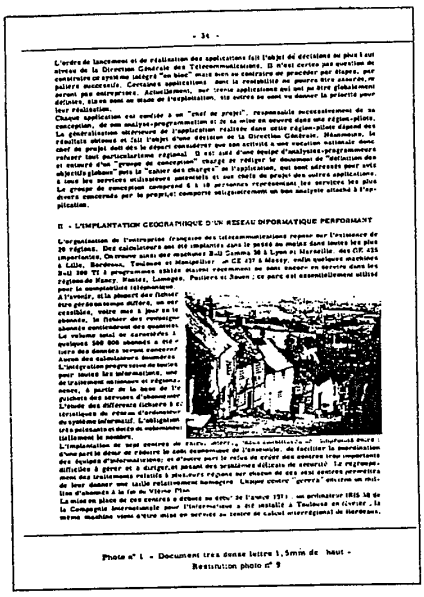


Figure 1. Test image for Mixed Color Facsimile simulation

Table 1. Simulation results of Mixed Color Facsimile compression

	JPEG only	Mixed color compression		
		with MH	with MMR	with JBIG
B & W doc. 200 dpi	519 Kbytes	76 Kbytes	48 Kbytes	45 Kbytes
Full color 200 dpi		81 Kbytes	81 Kbytes	81 Kbytes
Total	519 Kbytes	157 Kbytes	129 Kbytes	126 Kbytes

In order to apply this method to the scan and send type color facsimile system which scans and sends almost simultaneously, the following are required;

1) document syntax that can represent the structure of the area separated color images.
2) automatic area separation method.

Therefore, we studied document syntax and developed these items. A simulation system was constructed on a work-station for confirming the efficiency and applicability of the proposed method, This paper reports the structure of the document syntax, automatic area separation method and simulation results.

ISSN0097-0966X/94/2501-0887-$1.00 + .00 © 1994 SID

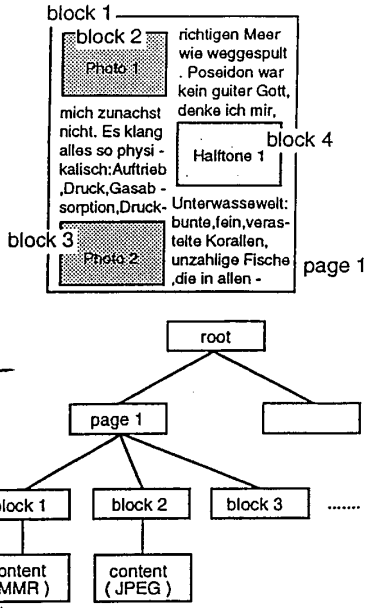


Figure 2. The structure of mixed color documents

DOCUMENT SYNTAX FOR MIXED COLOR FACSIMILE

For transmitting a mixed color facsimile document, a document syntax that represents the document structure is necessary. For this simulation system, we extended the Group 4 class 1 facsimile syntax. The Group 4 class 1 syntax is a subset of ODA and can be extended to support structured documents. In this syntax, only the layout structure is used and root, page and block are specified. This structure is shown in Figure 2. Contents used for this system are limited to the raster graphic contents of ODA.

An ASN.1(Abstract Syntax Notation 1) representation of this syntax is shown in Figure 3. Color related attributes that are used in this system are introduced from the ODA Colour extension and JPEG related attributes are newly added. Image encoding schemes used in the simulation system are JPEG for full color images and MMR for B&W images.

AUTOMATIC IMAGE AREA SEPARATION

For easy operation of the mixed color facsimile, an automatic image area separation technique is necessary. In this system, spatial frequency analysis using DCT(Discrete Cosine Transform) is used to distinguish character and photographic image area. Spatial frequency analysis is a well known method for this kind of process, but it is difficult to apply this method to scanned images whose resolutions range from 200 or 300 ppi(pel/inch) resolution because the differences between the DCT coefficients of these images are not clear. Therefore, we analyzed the spatial frequency property of these images using several subsample ratios from 1/1 to 1/3.

Statistical Characteristics of DCT Coefficients

We regard [g] as an (M x N) two dimensional image data matrix, and [G] as the two dimensional discrete cosine transformed data matrix of [g]. In this case, the element (u, v) of [G] is given as;

```
--- Layout Object Descriptor ---
Layout-Object-Descriptor      ::= SEQUENCE {
    object-type                   Layout-Object-Type,
    descriptor-body               Layout-Object-Descriptor-Body OPTIONAL }
Layout-Object-Type            ::= INTEGER {     document-layout-root   (0),
                                                page                   (2),
                                                block                  (4) }
Layout-Object-Descriptor-Body ::= SET {
    position                      [3]IMPLICIT Measure-Pair OPTIONAL,
    dimensions                    [4]IMPLICIT Dimension-Pair OPTIONAL,
    presentation-attributes       [6]IMPLICIT Presentation-Attributes OPTIONAL}
Measure-Pair                  ::=SEQUENCE {
    x-position                    [0] IMPLICIT INTEGER,
    y-position                    [0] IMPLICIT INTEGER}
Dimension-Pair                ::= SEQUENCE {
    horizontal                    [0]IMPLICIT INTEGER,
    vertical                      CHOICE {
        fixed                         [0]IMPLICIT INTEGER}}
Presentation-Attributes       ::= SET {
    raster-graphics-attributes    [1]IMPLICIT Raster-Graphics-Attributes OPTIONAL}
Raster-Graphics-Attributes    ::= SET {
    pel-transmission-density      [2]IMPLICIT Pel-Transmission-Density OPTIONAL}
Pel-Transmission-Density      ::= INTEGER {     p6 (200 dpi)           (1),
                                                p3 (400 dpi)           (4)}

--- Text Unit ---
Text-Unit                     ::= SEQUENCE {
    content-portion-attributes    Content-Portion-Attributes OPTIONAL,
    content-information            Content-Information                 }

Content-Portion-Attributes ::= SET {
    type-of-coding                Type-of-Coding       OPTIONAL,
    coding-attributes             CHOICE {
      raster-gr-coding-attributes   [2]IMPLICIT Raster-Gr-Coding-Attributes}
                                                                   OPTIONAL}
Raster-Gr-Coding-Attributes   ::= SET {
    number-of-pels-per-line       [0] IMPLICIT INTEGER                OPTIONAL,
    number-of-lines               [1] IMPLICIT INTEGER                OPTIONAL,
    subsampling                   [10]IMPLICIT Subsampling   OPTIONAL,
    jpeg-coding-mode              [11]IMPLICIT INTEGER { baseline (0)} OPTIONAL}
Type-of-Coding                ::= {0]IMPLICIT INTEGER{  t-6 (MMR)      (0),
                                                        t-81 (JPEG)    (16),
                                                        t-82 (JBIG)    (17)}
Subsampling                   ::= SEQUENCE {
    first-component               IMPLICIT Sub-Sample-Pair,
    second-component              IMPLICIT Sub-Sample-Pair,
    third-component               IMPLICIT Sub-Sample-Pair}
Sub-Sample-Pair               ::= SEQUENCE {
    horizontal                    INTEGER,
    vertical                      INTEGER}
Content-Information           ::= OCTET STRINGS { t-6 or t-81 or t-82 }
END
```

Figure 3. ASN.1 definition for Mixed Color Facsimile simulation system

$$G(u,v)=\frac{2c(u)c(v)}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} g(m,n)\cos\left(\frac{(2m+1)u\pi}{2M}\right)\cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

$$0\le u\le M-1,\quad 0\le v\le N-1,\qquad c(k)=\frac{1}{\sqrt{2}}\ (k=0),\quad =1\ (k\neq 0) \qquad (1)$$

This spatial frequency analysis applied to the luminance component of the color image obtained by color scanner. The CCITT test document No. 4 and photographic area of the Test chart No. 5 of The Society of the Electrophotography of Japan were used as sample images for character image and photographic image. Spatial frequency characteristics were calculated as follows. Absolute values of DCT coefficients G(u, v) were calculated for each 8 x 8 block of each subsampled image and averaged for the entire image area. The result was plotted for the order of (8*v + u). Figure 4 shows the result of the character image and figure 5 shows the result of the photographic image.

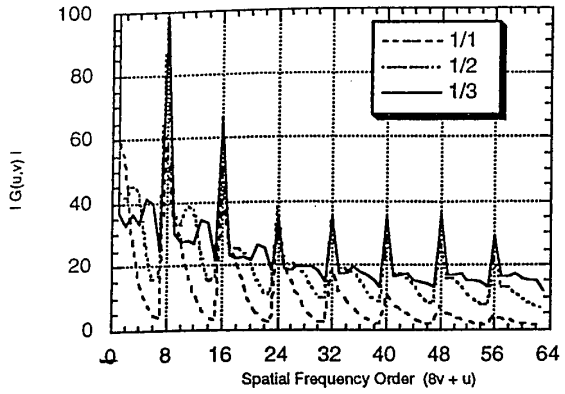


Figure 4. DCT coefficient characteristics of character image for different subsampling ratios.

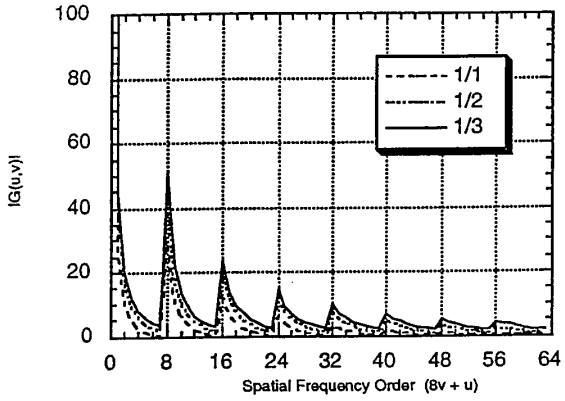


Figure 5. DCT coefficient characteristics of photographic image for different subsampling ratios.

These two figures show almost the same characteristics for 1/1 subsampling. There are, however, remarkable differences in high frequencies u and v with 1/2 and 1/3 subsampling. According to these results, character image areas and photographic image areas can be distinguish using 1/2 or 1/3 subsampling and DCT coefficient analysis.

Discrimination of photographic image and character image

In order to determine the discrimination function, the DCT coefficients matrix is divided to four groups: DC component, group A, group B, and group C, as show in Figure 6.

The dominant DCT coefficient groups for character images are the DC component and group C. DC component is influenced by background region of character documents which is generally white. Group C components correspond to the edge structure of character images. Group B components are also important for composing charter shapes, but they also change with photographic images.

The dominant DCT coefficient group for photographic images is group A, which corresponds to gradually changing tone areas. High frequency components such as group C have quite low levels. DC component is also influenced by the background of photographic image areas. Therefore, we selected the variables x, y for discrimination function as follows;

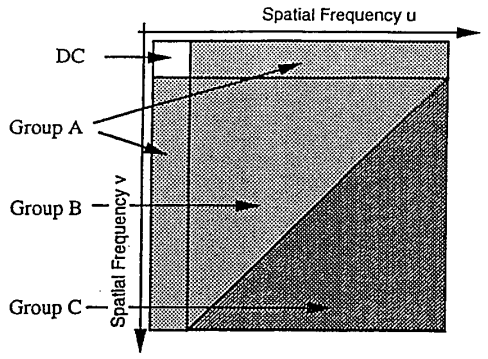


Figure 6. Separated DCT coefficient matrix

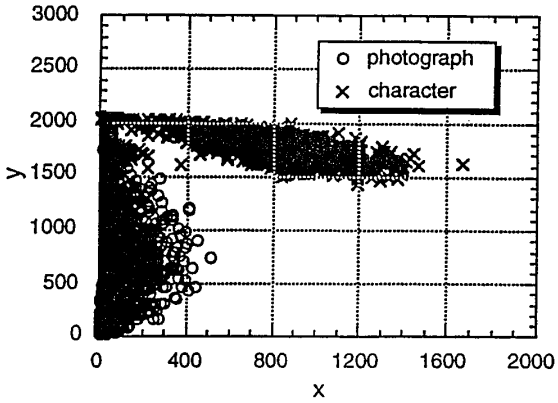


Figure 7. Distribution of (x, y) coordinate points

$$x = \sum_{u=0}^{8} \sum_{v=8-u}^{8} |G(u,v)|, \qquad y = G(0,0) \qquad (2)$$

The distribution of the x y coordinate of each block for character image areas and photographic image areas is shown in Figure 7.

From multi regression analysis, the discrimination function for character image areas and photographic image areas became as follows;

$$\begin{bmatrix} M(x_a)-M(y_a) \\ M(x_b)-M(y_b) \end{bmatrix}^T \begin{bmatrix} V(x_a)+V(y_a) & C(x_a y_a)+C(x_b y_b) \\ C(x_a y_a)+C(x_b y_b) & V(x_b)+V(y_b) \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\leq \begin{bmatrix} M(x_a)-M(y_a) \\ M(x_b)-M(y_b) \end{bmatrix}^T \begin{bmatrix} V(x_a)+V(y_a) & C(x_a y_a)+C(x_b y_b) \\ C(x_a y_a)+C(x_b y_b) & V(x_b)+V(y_b) \end{bmatrix}^{-1} \begin{bmatrix} M(x_a)+M(x_b) \\ M(y_a)+M(y_b) \end{bmatrix}$$

(3)

,where,

M(xa), M(ya) : the mean of x, y ,
V(xa), V(ya) : the variance of x, y ,
V(xa), V(ya) : the covariance between x and y , from photographic image;
M(xb), M(yb) : the mean of x, y ,
V(xb), V(yb) : the variance of x, y ,
V(xb), V(yb) : the covariance between x and y , from character image.

SIMULATION SYSTEM

At first, an original image stored in the file system is displayed as shown in Figure 9. On the display, it is possible to chose either automatic or manual image area separation mode. In the case of the automatic separation mode, the image is processed by the method described in Section 4. The discrimination function calculated in Section 4 is used. The result of discriminated result is displayed on the original image using rectangular area markers as shown in Figure 9. If the result is accurate, the image is divided into content blocks and encoded by JPEG and MMR coders. The SUI (Session User Information) that contains the structured image data for Mixed Color Facsimile communication is then assembled by the SUI encoder using the layout information and the coded image block data. In the receiving side, the transferred data is disassembled to yield the layout information and the coded block data. The full image is reconstructed using these data and displayed. This system can also print out the image through a digital color copying system (Canon CLC-500). The compressed image shown in Figure 9, occupies about 247 kbytes. This is about 60 % less than the JPEG only coded case (609kbyte). The printed example exhibits some image quality degradation, such as jerkiness in the character image area. This is because the character image area was binarized as a 200 dpi image. Higher resolution may be needed to avoid this degradation for binary images.

CONCLUSION

The mixed color facsimile and an automatic color image area separation method using DCT were proposed. The mixed color facsimile can reduce the amount of coded data by 60 % to 70 % from that needed by the JPEG only color facsimile. The automatic color image area separation method was applied to the test image, and its performance was confirmed.

REFERENCE

1. NTT "Color extension for G4 facsimile", CCITT COM VIII-80, January 1990.
2. M. Matsuki "Color extension for G4 facsimile", 6th International Workshop on Telematics, September 1991.
3. ITU-T Rec. T.410 series "Open Document Architecture (ODA) and Interchange Format".
4. ITU-T Rec. T.81 "Digital Compression and Coding of Continuous-tone Still Images" (JPEG).
5. ITU-T Rec. T.82 Progressive Bi-level Image Compression" (JBIG).

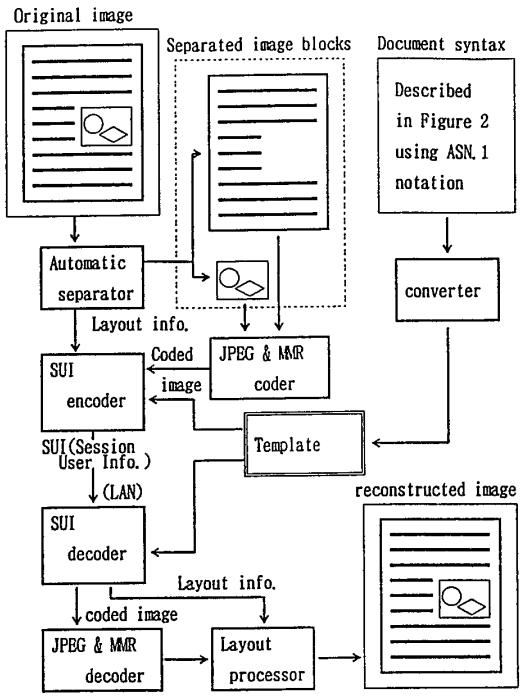


Figure 8. Process block diagram of Mixed Color Facsimile system

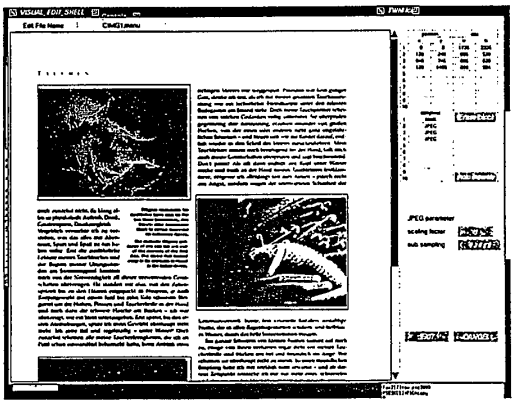


Figure 9. Display example of sending terminal

This figure is reproduced in color on page 996.

J. Physiol. (1985), **359**, *pp.* 381–400
With 11 *text-figures*
Printed in Great Britain

THE CONTRAST SENSITIVITY OF HUMAN COLOUR VISION TO RED–GREEN AND BLUE–YELLOW CHROMATIC GRATINGS

BY KATHY T. MULLEN

From the Physiological Laboratory, Cambridge CB2 3EG

(*Received 27 March 1984*)

SUMMARY

1. A method of producing red–green and blue–yellow sinusoidal chromatic gratings is used which permits the correction of all chromatic aberrations.

2. A quantitative criterion is adopted to choose the intensity match of the two colours in the stimulus: this is the intensity ratio at which contrast sensitivity for the chromatic grating differs most from the contrast sensitivity for a monochromatic luminance grating. Results show that this intensity match varies with spatial frequency and does not necessarily correspond to a luminance match between the colours.

3. Contrast sensitivities to the chromatic gratings at the criterion intensity match are measured as a function of spatial frequency, using field sizes ranging from 2 to 23 deg. Both blue–yellow and red–green contrast sensitivity functions have similar low-pass characteristics, with no low-frequency attenuation even at low frequencies below 0·1 cycles/deg. These functions indicate that the limiting acuities based on red–green and blue–yellow colour discriminations are similar at 11 or 12 cycles/deg.

4. Comparisons between contrast sensitivity functions for the chromatic and monochromatic gratings are made at the same mean luminances. Results show that, at low spatial frequencies below 0·5 cycles/deg, contrast sensitivity is greater to the chromatic gratings, consisting of two monochromatic gratings added in antiphase, than to either monochromatic grating alone. Above 0·5 cycles/deg, contrast sensitivity is greater to monochromatic than to chromatic gratings.

INTRODUCTION

The aim of this paper is to examine the spatial characteristics of human colour vision. For luminance vision this has been done by measuring a contrast sensitivity function: the ability of the visual system to detect luminance contrast at different spatial frequencies. The experiments described here aim to make comparable contrast sensitivity measurements for colour vision, by using grating stimuli which vary sinusoidally in colour.

A few previous studies have attempted to determine spatial sensitivity to red–green sinusoidal gratings, in which the two colours are matched in luminance to create an isoluminant stimulus (e.g. Schade, 1958; Van der Horst & Bouman, 1969; Granger & Heurtley, 1973; Kelly, 1983). Only one of these reports measurements using

blue–yellow sinusoidal stimuli (Van der Horst & Bouman, 1969). However, there are many difficulties associated with these investigations. First, the chromatic aberrations of the eye are likely to produce luminance artifacts in colour gratings at medium and high spatial frequencies. Transverse aberrations, or a chromatic difference of magnification, have not been corrected in previous isoluminant experiments. Corrections for longitudinal aberrations, or a chromatic difference of focus, have sometimes been made (Van der Horst & Bouman, 1969; Kelly, 1983). Secondly, a luminance match between the two colours in the stimulus has generally been made by using flicker photometry at one temporal and spatial frequency (Van der Horst & Bouman, 1969; Granger & Heurtley, 1973) and it has been assumed that this match is appropriate for all the other spatial and temporal frequencies used. However, red–green brightness matches may alter with temporal frequency (Ives, 1912; Börnstein & Marks, 1972), and so temporal and possibly spatial-frequency-dependent changes in brightness matches may have produced artifacts in previous isoluminant studies.

Thirdly, previous measurements have not extended to very low spatial frequencies and very few spatial cycles have been displayed at the lowest frequencies. A spatial cycle number below four or five is known to reduce sensitivity to luminance gratings (Findlay, 1969; Savoy & McCann, 1975). The lowest chromatic frequency that has been used while displaying four cycles is 0·4 cycles/deg (Granger & Heurtley, 1973) although often the lowest frequency measured with this cycle number has been higher at, for example, 1·4 cycles/deg (Van der Horst & Bouman, 1969). Furthermore, these latter measurements only extended down to spatial frequencies of 0·7 cycles/deg and for luminance gratings at comparable cycle numbers, low-frequency attenuation does not occur until below 0·5 cycles/deg (Howell & Hess, 1978). Thus, the previous studies have not satisfactorily investigated colour sensitivity to low spatial frequencies and the effects of reducing the spatial cycle number have not been distinguished from possible low-frequency attenuation below 0·5 cycles/deg. Finally, in previous investigations comparisons between colour and luminance sensitivities have not been attempted. This is partly because there is no adequate definition of colour contrast available which can be used for all colour combinations and does not depend on theoretical assumptions about post-receptoral cone interactions. Previous measures of colour sensitivity, such as purity (Van der Horst & Bouman, 1969) and wave-length discrimination, are difficult to relate to luminance contrast sensitivities.

The experiments described in this paper aim to overcome these problems in the following ways. (1) A different method of producing chromatic stimuli is used which permits correction of all chromatic aberrations. (2) Quantitative criteria are used to judge the most appropriate intensity match for creation of an optimum chromatic stimulus, and this match is adjusted separately at all spatial frequencies. (3) A very large field size is used which allows low spatial frequencies to be presented, without thresholds being affected by a low number of spatial cycles. (4) The stimulus is arranged so that the same contrast scale is used to determine thresholds for both chromatic and luminance gratings. This enables simple calculations to be made of the contrasts of the chromatic and luminance stimuli to individual cone types.

METHODS

The stimulus and procedure

A red–green chromatic grating was produced by displaying two gratings, each on Joyce display screens with white (P4) phosphors. These gratings were viewed through narrow band interference filters to produce their colour (Fig. 1). Interference filters with peak transmissions at 526 and 602 nm were chosen as these wave-lengths are at the peaks of both the human opponent colour spectral

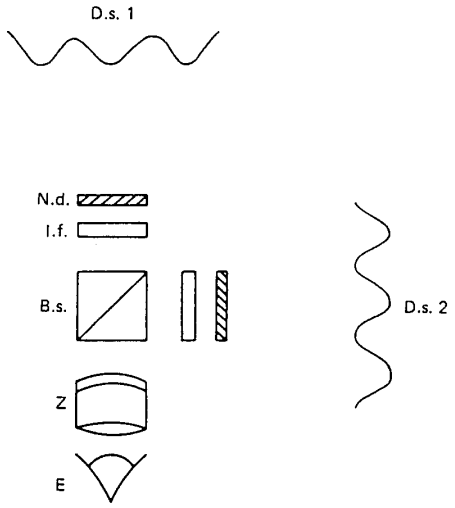


Fig. 1. A diagram of the experimental apparatus used to create the red–green and blue–yellow chromatic gratings. B.s., beam splitter; d.s. 1, d.s. 2, display screens Nos. 1 and 2; E, eye of observer; n.d., neutral density filter; Z, Zeiss telescope (×3); i.f., interference filter. Interference filters with peak wave-length transmissions of 602 and 526 nm were used to produce a red–green chromatic grating and filters with peaks at 470 and 577 nm were used for the blue–yellow grating.

sensitivity function (Sperling & Harwerth, 1971) and the chromatic response function of Hurvich & Jameson (1955). Thus, this red–green wave-length pair causes maximal modulation in the red–green chromatic response function but modulates the blue–yellow response function by only 12%. The two monochromatic gratings were combined optically 180 deg out of phase to form the composite chromatic grating. The chromatic grating patch was circular and ranged from 9·2 to 10·3 cm in diameter, depending on the correction made for the chromatic difference of magnification (described later). The remainder of the display screen was masked off with a diffuser; thus, at all contrasts used, the grating patch was set in a uniform surround of the same mean colour and reduced mean luminance. A fixation mark appeared at the centre of the chromatic grating. Viewing was monocular with a natural pupil and at a distance of 82 cm from each display screen. A Zeiss telescope (×3) could be placed directly in front of the eye. Viewing with the eye-piece close to the eye optically enlarged the grating and the field size, whereas viewing with the objective lens close to the eye optically reduces the image; it was thus equivalent to changing the viewing distance, and enabled the field size to be varied from 2·2 to 23·5 deg. The stimulus was phase reversed sinusoidally at 0·4 Hz.

The same method was used to produce a blue–yellow chromatic grating, but using interference filters with peak transmissions at 470 and 577 nm. 577 nm falls at the trough of the red–green

opponent spectral sensitivity function, and 470 nm is close to the blue peak. A filter transmitting light at the blue peak was not used because it severely reduced the mean luminance of the stimulus. This blue–yellow wave-length pair causes 74% modulation in the blue–yellow chromatic response function, but only 5% modulation in the red–green response function. Thus, the choice of the two wave-length pairs has been made on the basis of our knowledge of the post-receptoral colour opponent responses to different wave-lengths. As far as possible, chromatic gratings have been created which maximally stimulate one opponent colour system, and as such cause little modulation in the other opponent colour system.

Contrast of either component grating in the chromatic stimulus is defined by the usual formula:

$$C = \frac{I_{max} - I_{min}}{I_{max} + I_{min}},$$

where I_{max} and I_{min} are the peak and trough luminance values respectively of the monochromatic grating. The contrasts of the two component gratings were yoked together electronically, although their respective mean luminances may differ. Thus, $C_{526} = C_{602}$ and $C_{470} = C_{577}$ at all luminances. To find threshold, contrast is varied and at threshold the reciprocal contrast of either grating may be taken as the contrast sensitivity. Contrast output on the display screen was measured for a range of input contrasts using a UDT (United Detector Technology, model 40X) light-meter. Output contrast was linearly related to input contrast, and contrasts shown in the following experiments are the true, calibrated values.

Contrast output was also measured as a function of the spatial frequency on the display screen, using a psychophysical procedure which avoids the use of any additional optical apparatus with unknown modulation transfer characteristics. The subject set contrast thresholds for a range of gratings which consisted of pairs of stimuli identical in retinal spatial frequency (in cycles/deg) and retinal field size, but differing only in their screen spatial frequency (in cycles/cm) and viewing distance. Thus, any differences found between the thresholds for a pair of stimuli are likely to be due to the loss of contrast on the display screen at higher spatial frequencies. The results, shown in Fig. 2, reveal a non-linear relation between contrast output and screen spatial frequency: contrast output declines markedly above 0·4 cycles/cm and the loss is 40% at 2 cycles/cm. In the following experiments, screen spatial frequencies above 1·8 cycles/cm were not used. All contrast values quoted are of contrast output calibrated from the data of Fig. 2. The results of this psychophysical procedure agree well with results obtained from optical measurements of contrast loss for the same type of apparatus (Hess & Baker, 1984). Natural pupil sizes for the red–green stimuli were around 4 mm, and 6 mm for the blue–yellow stimuli. All mean luminances were measured using a calibrated SEI spot photometer.

Contrast thresholds were determined by a single staircase procedure (Cornsweet, 1962), begun at a randomly selected contrast above or below threshold. The grating was displayed continuously to increase the speed of threshold setting and to reduce considerably temporal transients. A mean of at least four thresholds was obtained for each plotted data point. The largest standard deviation of the thresholds is marked on each data curve. A 6809 Motorola microprocessor was used on-line to control the stimulus production and presentation, and data collection.

Three subjects were used in the experiments: K.T. (the author), R.M.C. and S.C.S. At least two subjects, and in some cases three, were used in each experiment. All subjects wore their normal correcting lenses, and performed normally on the Farnsworth–Munsell 100 hue test and the Ishihara test for colour blindness.

Correction of chromatic aberrations

This method of grating production has the advantage over the use of colour TV displays in that it allows the chromatic difference of focus and the chromatic difference of magnification of the eye and other optics to be corrected. The difference of focus may be corrected by placing a negative lens in the path of the shorter wave-length of the grating pair or a positive lens in the path of the longer wave-length, before the two component gratings are combined by the beam splitter.

It is also possible to measure the magnitude of this correction directly. The stimulus was arranged such that in the top half of the test patch one monochromatic square-wave component grating was displayed, whereas in the bottom half the other one appeared. The subject fixated on the longer-wave-length member of the pair (602 or 577 nm) with the help of a fixation mark. A series of negative correcting lenses was placed in front of the shorter-wave-length stimulus (470 or 526 nm)

until the subject saw this stimulus in sharpest focus simultaneously with the longer-wave-length grating. This method indicated that a correction of −1 D was required for the blue grating in the blue–yellow pair and a correction of −0·5 D was required for the green grating in the red–green pair. These values are close to previous calculations (see Wyszecki & Stiles, 1967) and were used in the present experiments.

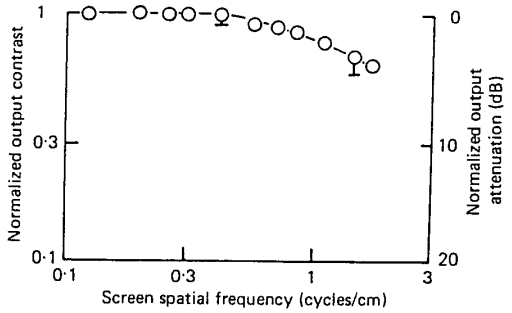


Fig. 2. Output contrast (c) normalized to contrast threshold as a function of screen spatial frequency (cycles/cm). A psychophysical method, described in the text, is used to calculate output contrast. Output contrast declines after 0·4 cycles/cm. Real contrast may be calculated from the curve by multiplying the uncalibrated input contrast by the normalized output contrast, or by adding the normalized output attenuation to the uncalibrated input attenuation. The smallest and largest standard deviations are shown. Attenuation (dB) = $20 \times \log 1/c$.

This empirical method of measuring the chromatic difference of focus is convenient to use since theoretical calculations become complex when the telescope is used to magnify or minify the stimulus, and will depend on the design of the telescope. When the telescope was used to magnify, very little correction was required for the short-wave-length gratings (−0·25 D for the 470 nm grating only). When the telescope was used to minify, much larger correcting lenses were needed, since for this reverse viewing condition a small difference of focus at the eye requires large correcting lenses at the eye-piece. A +3 D lens for the yellow grating in the blue–yellow stimulus, and a +2 D lens for the red grating in the red–green stimulus were found to be the best corrections.

The chromatic difference of magnification of the eye, and any additional optics in use, can be corrected by making independent adjustments to the spatial frequency of one of the component gratings. This was done by adjusting the X-gain on the appropriate display screen. Magnification differences are easily detected by displaying the two component gratings as square waves; overlap of adjacent bars produces a bright strip of a different colour which can be removed by adjusting the magnification of one grating.

Wave-length-dependent diffraction effects did not need correction as high frequencies, greater than 6 cycles/deg are not used (Van der Horst, de Weert & Bouman, 1967). While the chromatic aberrations are being corrected the subject's head is held in place using a dental bite bar and this line-up is maintained throughout the experiment. When the corrections have been made the gratings are displayed sinusoidally in space to produce a sinusoidal red–green or blue–yellow chromatic grating.

RESULTS

The removal of achromatic contrast

When creating stimuli which vary only in colour, an important problem is to establish the basis on which the intensities of the colours in the stimulus should be matched. Furthermore, it has frequently been assumed that a match made at one

spatial or temporal frequency will apply to all other frequencies. However, there is evidence to suggest that stimuli matched in luminance, for example by flicker photometry, will appear equally bright only under high spatial or high temporal frequency conditions, whereas under other low-frequency conditions luminance

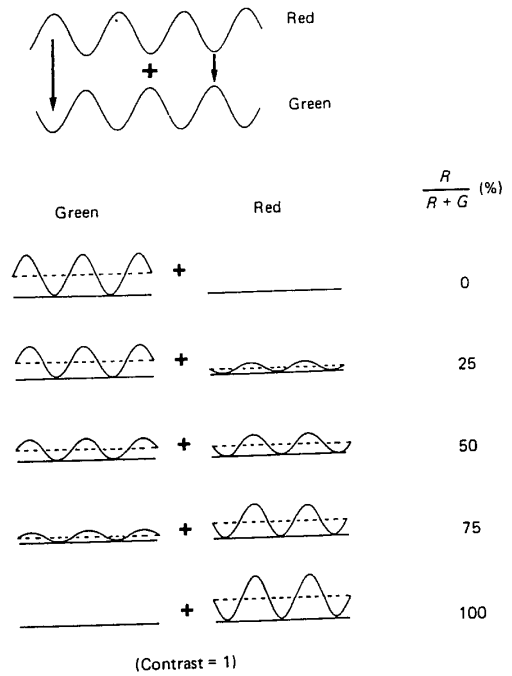


Fig. 3. A diagram of the luminance profiles across space of the red and green component gratings which are added 180 deg out of phase to produce a sinusoidal red–green chromatic stimulus. The ratio of red (R) to green (G) mean luminances in the chromatic grating is variable, and is expressed as the percentage of red light in the mixture. The mean luminance of the whole stimulus ($R+G$) is constant. The contrasts of the component red and green gratings are always equal and are at a value of 1 in this Figure. Contrast is varied to determine threshold. The same method is used to produce a blue–yellow chromatic grating, and the blue to yellow ratio is expressed as the percentage of yellow in the mixture.

matched stimuli will contain brightness differences (Ives, 1912; Börnstein & Marks, 1972; Myers, Ingling & Drum, 1973). Thus, there is a need to devise an appropriate criterion and a quantitative method for matching the intensity of the two colours in the stimulus which may be used at all spatial and temporal frequencies.

In this experiment, the ratio of the mean luminances of the two component gratings in the stimulus was varied over a wide range, and the subject's contrast sensitivity to the stimulus was measured at selected points. The criterion for the choice of the

intensity match was the luminance ratio at which the contrast sensitivity to the chromatic grating differs most from the contrast sensitivity to the monochromatic gratings. The method is illustrated for the red–green grating in Fig. 3. In this case, the ratio has been expressed as the percentage of red (R) in the red–green mixture. The range begins and ends with a red or green monochromatic stimulus that has luminance contrast but no colour contrast, and in the middle region the stimulus will

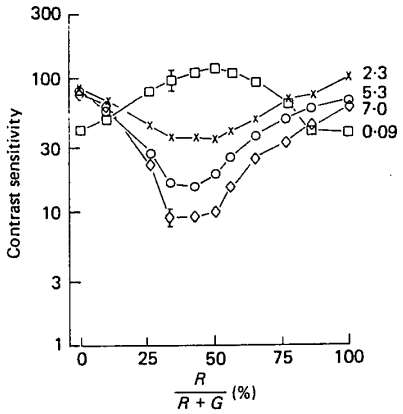


Fig. 4. Contrast sensitivity as a function of the red–green luminance ratio in the stimulus, expressed as the percentage of red in the mixture. Four spatial frequencies are shown (cycles/deg): ×, 2·3; ○, 5·3; ◇, 7·0 and □, 0·09. Vertical bars indicate ±1 S.D.. The subject is R.M.C.

have maximum colour contrast and minimal luminance contrast. Over-all there is no net change in the mean luminance of the composite stimulus; although R/G varies, $R+G$ was arranged to be at a constant photopic luminance (15 cd/m^2). The same method is used to vary the colour ratio in the blue–yellow stimulus. The ratio is expressed as the percentage of yellow in the mixture. The mean luminance of the composite stimulus ($B+Y$) remains constant at 2·1 cd/m^2.

Contrast sensitivity for one spatial frequency was measured at eleven or twelve percentages in the red–green or the blue–yellow range. The run was then repeated but beginning with the opposite colour in the range to avoid any effects due to chromatic adaptation. This was repeated for a range of spatial frequencies. Thus, the experiment examines the effect on detection of a monochromatic grating when a second grating of a different colour is added out of phase in various proportions. Typical results for the red–green grating are shown in Fig. 4, and for the blue–yellow grating in Fig. 5. The subject's contrast sensitivity is plotted as a function of the luminance ratio. The set of curves in each Figure represents a range of spatial frequencies.

The spatial frequency of the stimulus has a profound influence on the results. For low spatial frequencies (below 1 cycle/deg) the subject is less sensitive to the monochromatic conditions at either end, but as luminance contrast is reduced sensitivity *increases* reaching a maximum. However, for the higher spatial frequencies

the reverse occurs: the subject is most sensitive to the two monochromatic conditions, and in between sensitivity *decreases* reaching a minimum. Thus, under low spatial frequency conditions sensitivity is greatest when there are colour differences in the stimulus, whereas at higher frequencies sensitivity is greatest when the stimulus has only luminance contrast.

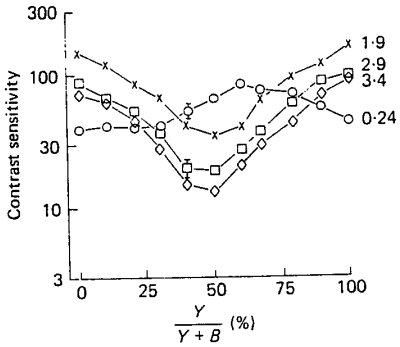


Fig. 5. Contrast sensitivity as a function of the blue–yellow luminance ratio in the stimulus, expressed as the percentage of yellow in the mixture. Four spatial frequencies are shown (cycles/deg): ×, 1·9; □, 2·9; ◇, 3·4 and ○, 0·24. The subject is K.T.

For the blue–yellow contrast sensitivities (Fig. 5) the minimum at high spatial frequencies[15] shifted relative to the maximum at low spatial frequencies. The low spatial frequency (0·24 cycles/deg) maximum occurs at 60 % yellow, or higher. At 1·9 cycles/deg a minimum occurs at 50 % yellow, and the remaining curves at 2·9 and 3·4 cycles/deg both have minima at 45 % yellow. All spatial frequencies in this Figure were displayed with the same field size (6·5 deg). Thus, for this subject (K.T.) as for others, there is a shift in the intensity match with spatial frequency of about fifteen percentage points. Most of this change occurs below 2 cycles/deg. Less blue is required at the low spatial frequency maxima than at the high spatial frequency minima, indicating that the effective intensity of the 470 nm wave-length is relatively lower at high frequencies. The red–green threshold data, shown in Fig. 4, are suggestive of a similar but much smaller shift. The low spatial frequency maxima occur at 55 % red, and the minima occur at 50 and 47 % red for 2 and 3 cycles/deg respectively. For other subjects a similar pattern occurs. This effect is not more than 7 %, but resembles the blue–yellow results in that relatively more of the shorter-wave-length (526 nm) light is required at the criterion match as spatial frequency increases up to 2 cycles/deg. Thus, for both red–green and blue–yellow stimuli a luminance match between colours, which occurs at 50 % red or 50 % yellow, does not predict the maxima or minima of contrast sensitivity.

It can also be seen from these results that the minima at high spatial frequencies become more sharply defined, making an accurate choice of intensity match more critical, since small differences in the match have quite large effects on sensitivity. These minima continue to increase in depth from 2 to 7 cycles/deg.

All subjects were asked to report any changes in the appearance of the gratings at threshold, at the different intensity ratios. The appearance varied from a homochromatic condition, where the bars appeared to be of a uniform colour but varying in brightness, to a heterochromatic condition where hue differences could be distinguished at threshold. At low spatial frequencies, colour differences could be

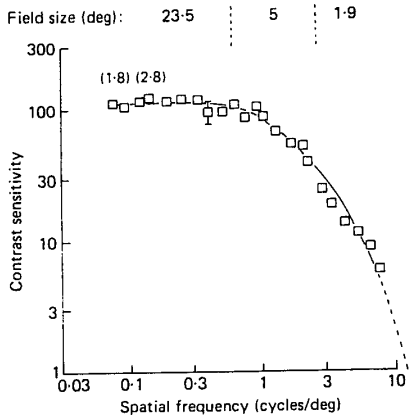


Fig. 6. Contrast sensitivity as a function of spatial frequency for a red–green grating (□: 526, 602 nm). Slightly different red–green ratios were used at different spatial frequencies to obtain the criterion intensity match of the two colours. The lowest numbers of spatial cycles displayed are indicated in parentheses. The continuous curve was fitted by eye. The method of extrapolation (dashed line) is described in the text. The subject is R.M.C. See also the upper curve of Fig. 7 for results of subject K.T.

detected at threshold for most of the intensity ratios. However, for the highest spatial frequencies used, such heterochromatic colour thresholds occurred at only 2 or 3 intensity ratios, and these always coincided with the minima of sensitivity. These observations strongly suggest that colour differences are detected at threshold at the intensity ratios which produce the maximal and minimal sensitivities. They also emphasize the need for an accurate, quantitative method of determining the match since, at high spatial frequencies, only a narrow range of intensity ratios produce colour detection thresholds. Furthermore, at the intensity ratios which occur at and around the maxima and minima of contrast sensitivity, the two colours in the grating appear as bars of equal brightness. Many subjects comment on the unusually vivid or 'fluorescent' appearance of the colours at these points.

The chromatic contrast sensitivity function (c.s.f.)

Measurements of the sensitivity of colour vision to different spatial frequencies can now be made using the criterion that the maxima and minima indicate the best intensity ratio for the two colours in the chromatic grating. For a range of spatial frequencies, results similar to those of Figs. 4 and 5 were obtained, and intensity ratios at the maxima and minima selected for determining the contrast sensitivities which are plotted in Figs. 6 and 7. The largest field size (23·5 deg) used in the experiment

Comparisons between colour and luminance c.s.f.s

The colour and luminance c.s.f.s differ in shape, but we do not know how their relative sensitivities compare. Comparisons of sensitivity are difficult since there is no adequate definition of colour contrast available which can be applied to all colour combinations, and does not depend on theoretical assumptions about post-receptoral

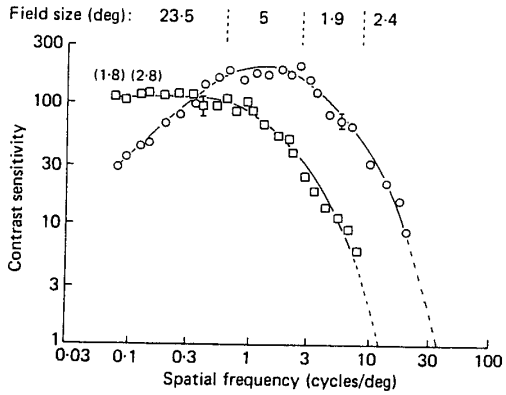


Fig. 8. Contrast sensitivity as a function of spatial frequency for the red–green grating (□; 526, 602 nm) and a green monochromatic grating (○; 526 nm). The data for the chromatic grating are taken from Fig. 6. The subject is R.M.C.

cone interactions. None of the previous measures of chromatic sensitivity, such as wave-length discrimination or purity, translate readily into the luminance domain. Measures of purity have resulted in the two component luminance gratings being presented at different contrasts, making comparisons with luminance sensitivity difficult. In the present experiments, the contrasts of the two component gratings are always held equal to each other, and at threshold the reciprocal contrast of either grating is taken as contrast sensitivity. Thus, as a working measure, the same contrast scale is used to determine detection thresholds for both the luminance and chromatic gratings. More direct and quantitative comparisons of sensitivity can also be made of the level of the cone responses since it is relatively simple to calculate the contrast of the luminance and chromatic gratings to each cone type.

The results shown in Figs. 4 and 5 give an initial indication of how contrast sensitivity changes as luminance contrast is removed and chromatic contrast is added to the stimulus. The present experiment extends these comparisons over the complete spatial range. The data for the chromatic gratings were taken from Figs. 6 and 7. Data for the luminance gratings were obtained by either using the pure green grating (0 % red condition) to make the red–green comparison, or using the pure yellow grating (100 % yellow condition) to make the blue–yellow comparison. Luminance and chromatic comparisons were each made at the same mean luminances. The choice of monochromatic grating is not important since Van Nes & Bouman (1967) have shown that the wave-length of a monochromatic luminance grating does not affect

enabled frequencies as low as 0·17 cycles/deg to be displayed with over 4 cycles present. Thus, low spatial frequency sensitivity could be assessed without being affected by a reduced cycle number, since if more than four spatial cycles are present contrast sensitivity is independent of the cycle number and the field size (Howell & Hess, 1978).

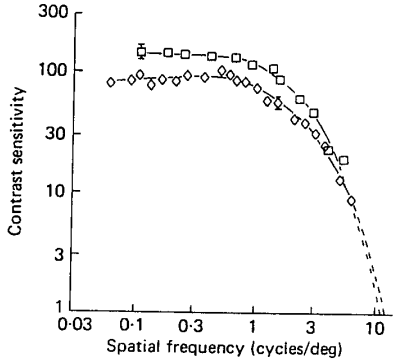


Fig. 7. Contrast sensitivities as a function of spatial frequency for a blue–yellow grating (◇; 470, 577 nm) and a red–green grating (□; 602, 526 nm), both for subject K.T. Different blue–yellow ratios were used at different spatial frequencies to obtain the criterion intensity match of the two colours. Slightly different red–green ratios were also required for the criterion match. The continuous curve was fitted by eye. The method of extrapolation (dashed line) is described in the text.

The results obtained using red–green gratings are shown in Fig. 6 for R.M.C. and in the upper curve of Fig. 7 for K.T., the blue–yellow results for K.T. are shown in Fig. 7. Sensitivities to both blue–yellow and red–green stimuli have low-pass characteristics, with no decline in sensitivity for spatial frequencies below 0·1 cycles/deg. Previous declines found (e.g. Kelly, 1983) may have been due to the low number of cycles displayed.

Sensitivity to the red–green and blue–yellow stimuli declines at spatial frequencies above 0·8 cycles/deg. Sensitivity to the red–green medium and higher spatial frequencies is lower than has been previously reported and by extrapolation, red–green chromatic resolution fails at 11–12 cycles/deg for R.M.C. and K.T. (The method of extrapolation is described later.) Previously, resolutions above 25 cycles/deg have been suggested. Resolution of the blue–yellow grating also fails at around 11 cycles/deg for both subjects K.T. and S.C.S. (no Figure). This compares with an acuity of above 20 cycles/deg, obtained using blue–yellow sine-wave stimuli (Van der Horst & Bouman, 1969). These chromatic acuity values are investigated more fully in a later section.

Fig. 7 shows a comparison between the red–green and blue–yellow sensitivities obtained from the same subject (K.T.). The two c.s.f.s are remarkably similar and have much the same high spatial frequency decline. The only significant difference occurs in the low spatial frequency region where the blue–yellow sensitivity is consistently about 0·15–0·2 log units lower.

contrast sensitivity provided the stimuli have the same mean luminance. The results for the comparison between sensitivities to the red–green chromatic grating and the green monochromatic grating are shown in Fig. 8. The blue–yellow chromatic and yellow monochromatic comparisons are shown in Fig. 9.

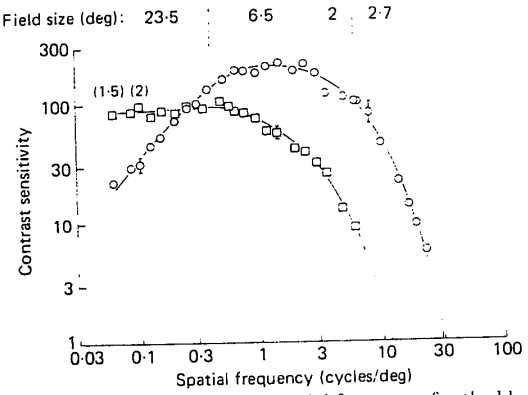


Fig. 9. Contrast sensitivity as a function of spatial frequency for the blue–yellow grating (□; 470, 577 nm) and a yellow monochromatic grating (○; 577 nm). The data for the chromatic grating are taken from Fig. 7. The subject is K.T.

The results show that the contrast sensitivity to both red–green and blue–yellow gratings is greatest below 1 cycle/deg, whereas luminance contrast sensitivity peaks at 0·8–4 cycles/deg. For the low spatial frequencies, the combination of the red and green monochromatic gratings in antiphase can be seen when neither grating can be seen alone. This difference in contrast sensitivity reaches 0·6 log units and may increase at even lower spatial frequencies. Results obtained on another subject (K.T.) are very similar. The same effect occurs for the blue–yellow stimuli. For low spatial frequencies, contrast sensitivity to the combination of monochromatic gratings in antiphase is greater than to the monochromatic grating alone. This difference reaches 0·5 log units at 0·1 cycles/deg. For another subject (S.C.S.) the difference was slightly less (0·4 log units). Above cross-over points at 0·3–0·5 cycles/deg for all subjects, contrast sensitivity becomes greatest to the monochromatic stimuli, and it is luminance vision which has the higher acuity.

Comparisons of chromatic and luminance acuity

Previous studies using isoluminant techniques have produced a wide range of values for chromatic acuity. In most studies, extrapolations have to be made by eye from threshold measurements obtained at lower spatial frequencies. Such procedures, using purity as the measure of chromatic sensitivity suggest acuity values for red–green gratings that range from 25–30 cycles/deg (Van der Horst & Bouman, 1969) to 50 cycles/deg and equal to luminance acuity (Schade, 1958). Two studies which include measurements made using blue–yellow sine or square-wave stimuli suggest

an acuity greater than 20 cycles/deg (Van der Horst *et al.* 1967; Van der Horst & Bouman, 1969). Studies which have attempted to measure acuity using isoluminant sine- or square-wave gratings of variable wave-lengths have also reported a similar range of acuity values from 20 to 30 cycles/deg (Hilz, Hupperman & Cavonius, 1974).

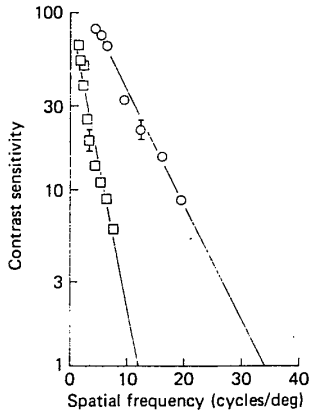


Fig. 10. Contrast sensitivity as a function of spatial frequency, plotted on semilogarithmic coordinates. The data for red–green gratings (□: 602, 526 nm) and green monochromatic gratings (○, 526 nm) are taken from Fig. 8. Linear regression lines are fitted to the data and extrapolated to a contrast sensitivity of 1 (100% contrast) to indicate acuity. Low spatial frequency data have been omitted (see text for further details). The subject is R.M.C.

and bar frequencies of 46 cycles/deg reported to equal luminance acuity under similar conditions (Cavonius & Schumacher, 1966). The purpose of the following calculations is to make accurate predictions of colour and luminance acuity on the basis of the new contrast sensitivity measurements obtained here.

The high spatial frequency data points for the luminance and chromatic gratings were replotted on semilogarithmic coordinates. All the data points which occur after the peak sensitivity of the colour or luminance contrast sensitivity functions are included in the plot. In effect, the medium and high spatial frequency points that occur at or below a contrast sensitivity of 100 were included. A linear regression line was fitted to each function and extrapolated to a contrast of 100% (contrast sensitivity = 1) to predict acuity.

Results for red–green stimuli are shown in Fig. 10 and the blue–yellow results in Fig. 11. Visual inspection reveals that the regression lines fit the data points well. Red–green chromatic acuity is 11–12 cycles/deg, compared to the luminance acuity of 34–36 cycles/deg at the same mean luminance for subjects R.M.C. and K.T. Blue–yellow chromatic acuity is around 11 cycles/deg, closely resembling red–green acuity, compared to the luminance acuity of 32–33 cycles/deg, for subjects K.T. and S.C.S.

Luminance acuity is lower than might be expected. This is probably due to the

relatively low mean luminance of the stimuli which will reduce sensitivity to very high spatial frequencies. However, comparisons with the results of previous chromatic studies can be made since equivalent or higher luminances have been used in the present experiments.

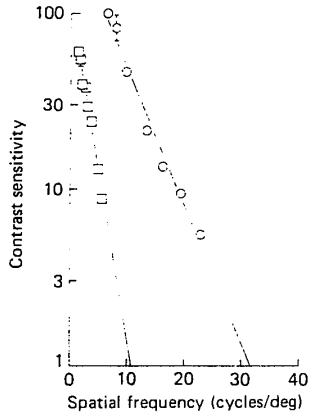


Fig. 11. Contrast sensitivity as a function of spatial frequency, plotted on semilogarithmic coordinates. The data for blue–yellow gratings (□: 470, 577 nm) and yellow monochromatic gratings (○, 577 nm) are taken from Fig. 9. Linear regression lines are fitted to the data and extrapolated to a contrast sensitivity of 1 (100% contrast) to indicate acuity. Low spatial frequency data have been omitted (see text for further details). The subject is K.T.

Thus, these results indicate that chromatic acuity, based on hue discriminations of sinusoidal chromatic gratings, is lower than previously thought at 11–12 cycles/deg for both the red–green and blue–yellow stimuli. Possible explanations for the higher sensitivities and acuities found in previous studies are considered in the Discussion.

Note on colour appearance

At suprathreshold levels these purely chromatic sine-wave gratings are square wave in appearance. For example, no intermediary shades of yellow are seen between the red and green peaks and little variation occurs in the appearance of these colours within each bar. A similar effect occurs for the blue–yellow stimulus, where no intermediary blue–whites are seen. The unexpected absence of yellow between regions of red and green, and the absence of other such 'transition' colours, has been commented on before, both in the spectrum (von Helmholtz, 1909), and using overlapping linear ramps of red and green (Campbell, 1983). Below about 0·3 cycles/deg, this effect disappears and the chromatic gratings become more sinusoidal in appearance.

DISCUSSION

These experiments have revealed a shift with spatial frequency in the intensity match which produces the maximum change in contrast sensitivity. The shift is most prominent for blue–yellow gratings and shows that the effectiveness of blue light

relative to yellow in the match, decreases as spatial frequency increases up to 2 cycles/deg. There is also a suggestion of a similar but smaller shift in the red–green match, where the effectiveness of green light decreases relative to red at the higher spatial frequencies. The question arises as to what causes these changes in match point. Wave-length-dependent diffraction effects are unlikely since the shift occurs at relatively low spatial frequencies, below 6 cycles/deg. Also, diffraction would cause a relative decrease in the contrast of the red or yellow grating, and so would produce a shift in the opposite direction at higher spatial frequencies. Small differences in focus between the two colours due to longitudinal chromatic aberrations might cause an apparent shift in an intensity match, by reducing the contrast of one colour. However, in the present experiments chromatic aberrations have been corrected, and a considerable change in match still occurs for blue–yellow stimuli at very low spatial frequencies below 1–2 cycles/deg. Any small residual differences in focus between the two colours are unlikely to affect thresholds at these low spatial frequencies (Campbell & Green, 1965).

Another possible explanation is that blue cones or rods contribute to the match under low spatial frequency conditions, but not at higher spatial frequencies, therefore decreasing the effectiveness of short wave-length light in the match at these higher frequencies. It is known that the sensitivity of the 'isolated' blue system decreases above 1–2 cycles/deg and is considerably reduced by 5–6 cycles/deg (Kelly, 1974; Green, 1972), which is broadly compatible with the shift occurring at low spatial frequencies. The fact that the shift is considerably greater for the blue–yellow match than for the red–green one is compatible with a blue-sensitive mechanism being involved. Rod sensitivity also declines above 1 cycle/deg (Green, 1972). However, rods are unlikely to contribute to threshold since, at threshold, different colours can be seen in the stimulus. These results suggest that spatial frequency influences brightness perception; and are compatible with other evidence which shows that brightness differences are not always predicted by the standard V_λ luminosity function (Ives, 1912; Börnstein & Marks, 1972; Myers *et al.* 1973).

These results have shown that acuities for the red–green and blue–yellow gratings are very similar, namely 10–12 cycles/deg. Although our knowledge of post-receptoral colour processing is very limiting, the wave-length pairs for the two gratings were chosen so as to optimally stimulate either the red–green or the blue–yellow opponent colour system, and each causes very little response in the opposite opponent system (see Methods). Thus, it is likely that the detection of the red–green and blue–yellow gratings is by the red–green and blue–yellow opponent colour systems respectively. It is interesting that the red–green colour acuity is so low in view of the dense distribution of red and green cone types in the retina. The acuity for the blue–yellow grating agrees well with recent estimates of the acuity of the 'isolated' blue mechanism, also at 10–14 cycles/deg (Stromeyer, Kranda & Sternheim, 1978; Williams, Collier & Thompson, 1983). Thus, the results may suggest that the sparse distribution of blue cones in the retina is not the only factor limiting blue–yellow grating acuity. Previous measurements have suggested much higher chromatic acuity values ranging from 20 to 30 cycles/deg to normal luminance acuities. The methods used here allow accurate measurements of sensitivity to chromatic high spatial frequencies to be made since a quantitative way of making an intensity match has been adopted: the accuracy of this match is shown to be most important at high

spatial frequencies. Furthermore, corrections have been made for both types of chromatic aberration, reducing or eliminating luminance artifacts in the stimulus. In the experiments, the subjects could all detect the colour differences in the matched stimulus at threshold, at all spatial frequencies measured, suggesting that these thresholds are based on colour discriminations.

Reports by some other authors suggest that previous measurements of sensitivity to medium and high spatial frequency chromatic gratings are not based on the perception of colour differences. For example, Granger & Heurtley (1973) found that colour differences in the stimulus at threshold disappear at spatial frequencies above 3 cycles/deg, and that the remaining brightness differences could not be nulled by readjusting the colour match. Such effects might be explained if the medium and high spatial frequency thresholds were based on luminance artifacts in the stimulus produced by chromatic aberrations. Cavonius & Schumacher (1966), who measured acuities to chromatic gratings, did not look for colour differences in the stimulus but reported a wave-length discrimination function at 30 cycles/deg which is very unlikely to be based on hue discriminations. Another possibility which should be considered in this case is that the spectral sensitivity of the achromatic detecting mechanism changes at spatial frequencies greater than those used in the present experiment introducing brightness differences into the stimulus. If two achromatic detecting mechanisms were available then brightness differences could not be nulled simply by readjusting the brightness match. Further experiments eliminating all luminance artifacts at spatial frequencies above 7 cycles/deg are in progress to test these possibilities.

In the experiments described here, comparisons have been made between contrast sensitivities to luminance and chromatic gratings. Although contrast sensitivity to monochromatic gratings does not change with the wave-length (colour) of the stimulus, providing the mean luminance is constant (Van Nes & Bouman, 1967), the over-all contrast sensitivity to the chromatic gratings will depend on the particular colour pairs which they contain. Thus, any comparisons of sensitivity to luminance and chromatic gratings will be influenced by the colours of the pairs in the chromatic stimulus. For the comparisons made here, wave-lengths were chosen to coincide with the peaks of the opponent colour spectral sensitivity function and the chromatic response function (see Methods), and so the over-all contrast sensitivity to the chromatic gratings is unlikely to be greatly increased, but may be decreased, by using different wave-lengths. Also, measurements made of modulation sensitivities to different wave-length combinations (Butler & Riggs, 1978) confirm that sensitivity is relatively high to the colour pairs used here.

Both red and green gratings in the red–green stimulus will stimulate both medium- and long-wave-length cone types and even at isoluminance the stimulus will contain intensity differences to individual cone types. Thus, comparisons between the luminance and colour c.s.f.s can also be made in terms of their cone contrasts. Calculations have been made in the Appendix which show that, at the red–green ratio used for subject R.M.C. in the low spatial frequency chromatic grating, the contrast of this grating to a mechanism with the spectral sensitivity of long-wave-length cones is 18% of the contrast of either component grating. For a mechanism with the spectral sensitivity of medium-wave-length cones, the contrast of the chromatic grating is 39%

of the contrast of either component grating. These values at the criterion red–green match for another subject (K.T.) are also given in the Appendix.

The comparisons of contrast sensitivities have revealed that at low spatial frequencies the two monochromatic gratings combined in antiphase can be seen when neither grating can be seen alone. For example, at the lowest spatial frequency contrast sensitivity to the red–green grating is 3·8 times greater than to the green grating presented alone (subject R.M.C., Fig. 8). However, when considered in terms of cone contrasts, this effect is considerably greater. The modulations of the long-wave-length cones which can be detected in the chromatic condition are 21 times smaller than those which can be detected for the monochromatic grating presented alone. For medium-wave-length cones, modulations 10 times smaller can be detected when the stimulus is in the chromatic (antiphase) condition than when either monochromatic stimulus is presented alone. Thus, at low spatial frequencies a chromatic grating can be detected on the basis of considerably smaller receptor modulations than can a luminance grating. This interesting effect is presumably mediated by the post-receptoral extraction of colour opponent signals, involving the combination of different cone outputs.

Finally, the psychophysical results reported here are relevant to the neurophysiology of primate colour vision. The evidence has shown that the relative sensitivities of the visual system to colour and luminance contrast change with spatial frequency. Since colour opponent cells are likely to respond to both colour and luminance contrast (Ingling & Drum, 1973), it can be predicted that the relative sensitivity of these single cells to colour and luminance contrast is spatial frequency dependent. Thus, these psychophysical results emphasize the importance in future neurophysiological studies of considering spatial variables when determining the colour and luminance contrast sensitivities and the spectral sensitivities of single cells.

APPENDIX

The following calculations are of the effective contrast (C'_c) of a chromatic grating, composed of two monochromatic gratings added in antiphase, for a single cone type.

The quantal intensity profile (I_c) of the chromatic grating is described by:

$$I_c = M_1\alpha_1 + M_2\alpha_2 + (a_1\alpha_1 - a_2\alpha_2)\sin\omega x,$$

where $\frac{\omega}{2\pi}$ is its spatial frequency and x is space. The contrast of the grating is:

$$C'_c = \frac{a_1\alpha_1 - a_2\alpha_2}{M_1\alpha_1 + M_2\alpha_2},$$

where: 1, 2 are subscripts denoting the wave-lengths of the component gratings: M_1, M_2 are the mean quantal intensities of each component grating: a_1, a_2 are the amplitudes of each component grating: α, β denote the spectral sensitivity weightings for the wave-lengths of the two component gratings for long (α)- and medium (β)-wave-length cone types.

If the contrasts of the two component gratings are equal and at a value C'

$$M_1 = a_1/C',$$
$$M_2 = a_2/C'$$

and

$$C_c = \frac{a_1\alpha_1 - a_2\alpha_2}{a_1\alpha_1 + a_2\alpha_2} \times C. \quad (1)$$

If the ratio of the luminance of component grating No. 1 to component grating No. 2 is L, their quantal intensities are equated by:

$$a_1 V_1 = L a_2 V_2,$$

or

$$a_1 = L a_2 V, \quad (2)$$

where $V = V_2/V_1$; V_1, V_2 are the standard V_λ luminous efficiency weightings of the component wave-lengths.

Substituting eqn. (2) in eqn. (1):

$$C_c = \frac{LV\alpha_1 - \alpha_2}{LV\alpha_1 + \alpha_2} \times C. \quad (3)$$

For the red–green chromatic grating used in the present experiments, wave-length No. 1 is 526 nm and wave-length No. 2 is 602 nm

$$V_{526} = 0{\cdot}8012,$$

$$V_{602} = 0{\cdot}6054.$$

Therefore, $V = 0{\cdot}7556$.

Cone spectral sensitivities may be taken from the Smith & Pokorny (1975) cone sensitivity functions, based on colour matching data (see Boynton, 1979).

For long-wave-length cones (α)

$$\alpha_{526} = 0{\cdot}4526,$$
$$\alpha_{602} = 0{\cdot}4905.$$

For medium-wave-length cones (β)

$$\beta_{526} = 0{\cdot}3484,$$
$$\beta_{602} = 0{\cdot}1149.$$

The data in Fig. 4 for subject R.M.C. show that the criterion intensity match at low spatial frequencies is at 50 % red. Thus the green to red luminance ratio (L) = 1.

Using these values in eqn. (3) gives:

$C_c = -0{\cdot}1784 \times C$ for long-wave-length cones, or 18 % of C:

and

$C_c = +0{\cdot}3923 \times C$ for medium-wave-length cones, or 39 % of C.

For subject K.T., the intensity match at low spatial frequencies is at 55 % red. Thus, the green to red luminance ratio (L) = 0·8182.

Using these values in eqn. (3) gives:

$C_c = -0{\cdot}2735 \times C$ for long-wave-length cones, or 27 % of C:

and

$C_c = +0{\cdot}3043 \times C$ for medium-wave-length cones, or 30 % of C.

I would like to thank Professor H. B. Barlow and Professor F. W. Campbell for their interest in this work, and I am grateful to Professor Barlow for providing laboratory space and equipment. I am also indebted to my subjects (Roselyn M. Cummings and Sally Cline-Smith) for their patience and co-operation. I am in receipt of a Research Fellowship from New Hall, Cambridge.

REFERENCES

BÖRNSTEIN, M. N. & MARKS, L. E. (1972). Photopic luminosity measured by the method of critical frequency. *Vision Research* **12**, 2023–2034.

BOYNTON, R. M. (1979). *Human colour vision*. New York: Holt, Rinehart and Winston.

BUTLER, T. W. & RIGGS, L. A. (1978). Colour differences scaled by chromatic modulation sensitivity functions. *Vision Research* **18**, 1407–1416.

CAMPBELL, F. W. (1983). Cambridge Colour Contributions. In *Colour Vision: Physiology and Psychophysics*, ed. MOLLON, J. D. & SHARPE, L. T. pp. xxi–xxv. London: Academic Press.

CAMPBELL, F. W. & GREEN, D. G. (1965). Optical and retinal factors affecting visual resolution. *Journal of Physiology* **181**, 576–593.

CAVONIUS, C. R. & SCHUMACHER, A. W. (1966). Human visual acuity measured with coloured test objects. *Science* **152**, 1276–1277.

CORNSWEET, T. N. (1962). The staircase method in psychophysics. *American Journal of Psychology* **75**, 485–491.

FINDLAY, J. M. (1969). Spatial integration effect in visual acuity. *Vision Research* **9**, 157–166.

GRANGER, E. M. & HEURTLEY, J. C. (1973). Visual chromaticity modulation transfer function. *Journal of the Optical Society of America* **63**, No. 9, 73–74.

GREEN, D. (1972). Visual acuity in the blue cone monochromat. *Journal of Physiology* **222**, 419–426.

HESS, R. F. & BAKER JR, C. L. (1984). The human pattern evoked electroretinogram. *Journal of Neurophysiology* **51**, 939–951.

HILZ, R. L., HUPPMAN, G. & CAVONIUS, C. R. (1974). Influence of luminance contrast on hue discrimination. *Journal of the Optical Society of America* **64**, 763–766.

HOWELL, E. R. & HESS, R. F. (1978). The functional area for summation to threshold for sinusoidal gratings. *Vision Research* **18**, 369–374.

HURVICH, L. M. & JAMESON, D. (1955). Some quantitative aspects of an opponent-colours theory. II. *Journal of the Optical Society of America* **45**, 602–616.

INGLING JR, C. R. & DRUM, B. A. (1973). Retinal receptive fields: correlations between psychophysics and electrophysiology. *Vision Research* **13**, 1151–1163.

IVES, F. (1912). Studies of photometry of light of different colours. *Philosophical Magazine* **24**, 149–188 and 352–370.

KELLY, D. H. (1974). Spatio-temporal frequency characteristics of colour vision mechanism. *Journal of the Optical Society of America* **64**, No. 7, 983–990.

KELLY, D. H. (1983). Spatiotemporal variation of chromatic and achromatic contrast thresholds. *Journal of the Optical Society of America* **73**, 742–750.

MYERS, K. J., INGLING, C. R. & DRUM, B. A. (1973). Brightness additivity for a grating target. *Vision Research* **13**, 1165–1173.

SAVOY, R. L. & MCCANN, J. J. (1975). Visibility of low spatial frequency sine-wave targets: dependence on number of cycles. *Journal of Optical Society of America* **65**, No. 3, 343–350.

SCHADE, O. (1958). On the quality of color-television images and the perception of colour detail. *Journal of the Society of Motion Pictures and Television Engineers* **67**, No. 12, 801–819.

SMITH, V. C. & POKORNY, J. (1975). Spectral sensitivity of the foveal cone pigments between 400 and 500 nm. *Vision Research* **15**, 161–171.

SPERLING, H. G. & HARWERTH, R. S. (1971). Red–green cone interactions in increment thresholds of spectral sensitivity of primates. *Science* **172**, 180–184.

STROMEYER, C. F., KRANDA, K. & STERNHEIM, C. E. (1978). Selective chromatic adaptation at different spatial frequencies. *Vision Research* **18**, 427–438.

VAN DER HORST, G. J. C. & BOUMAN, M. A. (1969). Spatio-temporal chromaticity discrimination. *Journal of the Optical Society of America* **59**, No. 11, 1482–1488.

VAN DER HORST, G. J. C., DE WEERT, C. M. M. & BOUMAN, M. A. (1967). Transfer of spatial chromaticity contrast at threshold in the human eye. *Journal of the Optical Society of America* **57**, 260–1266.

VAN NES, F. L. & BOUMAN, M. A. (1967). Spatial modulation transfer in the human eye. *Journal of the Optical Society of America* **57**, 401–406.

VON HELMHOLTZ, H. (1909). *Handbook of Physiological Optics*, vol. II. Translated from the 3rd German edition in *Helmholtz's Treatise on Physiological Optics* (1962). Vols. I and II. ed. JAMES, P. C. New York: Southall, Dover Publications Inc.

WILLIAMS, D. R., COLLIER, R. J. & THOMPSON, B. J. (1983). Spatial resolution of the short wavelength mechanism. In *Colour Vision: Physiology and Psychophysics*, ed. MOLLON, J. D. & SHARPE, L. T. Academic Press.

WYSZECKI, G. & STILES, W. S. (1967). *Colour Science*, p. 212. New York: John Wiley and Sons.

Vision Res. Vol. 31, No. 7/8, pp. 1399–1415, 1991
Printed in Great Britain.

0042-6989 91 $3.00 + 0.00

OBJECT SPATIAL FREQUENCIES, RETINAL SPATIAL FREQUENCIES, NOISE, AND THE EFFICIENCY OF LETTER DISCRIMINATION

DAVID H. PARISH and GEORGE SPERLING*

Human Information Processing Laboratory, Department of Psychology and Center for Neural Sciences, New York University, NY 10003, U.S.A.

(Received 7 July 1988; in revised form 2 June 1990)

Abstract—To determine which spatial frequencies are most effective for letter identification, and whether this is because letters are objectively more discriminable in these frequency bands or because can utilize the information more efficiently, we studied the 26 upper-case letters of English. Six two-octave wide filters were used to produce spatially filtered letters with 2D-mean frequencies ranging from 0.4 to 20 cycles per letter height. Subjects attempted to identify filtered letters in the presence of identically filtered, added Gaussian noise. The percent of correct letter identifications vs *s/n* (the root-mean-square ratio of signal to noise power) was determined for each band at four viewing distances ranging over 32:1. Object spatial frequency band and *s/n* determine *presence of information* in the stimulus; viewing distance determines retinal spatial frequency, and affects only *ability to utilize*. Viewing distance had no effect upon letter discriminability: object spatial frequency, not retinal spatial frequency, determined discriminability. To determine discrimination efficiency, we compared human discrimination to an ideal discriminator. For our two-octave wide bands, *s/n* performance of humans and of the ideal detector improved with frequency mainly because linear bandwidth increased as a function of frequency. Relative to the ideal detector, human efficiency was 0 in the lowest frequency bands, reached a maximum of 0.42 at 1.5 cycles per object and dropped to about 0.104 in the highest band. Thus, our subjects best extract upper-case letter information from spatial frequencies of 1.5 cycles per object height, and they can extract it with equal efficiency over a 32:1 range of retinal frequencies, from 0.074 to more than 2.3 cycles per degree of visual angle.

Spatial filtering Scale invariance Psychophysics Contrast sensitivity Acuity

INTRODUCTION

Characterizing objects

When we view objects, what range of spatial frequencies is critical for recognition, and how is our visual system adapted to perceive these frequencies? Ginsburg (1978, 1980) was among the first to investigate this problem by means of spatial bandpass filtered images of faces and lowpass filtered images of letters. He noted the lowest frequency band for faces and the cutoff frequency for letters at which the images seemed to him to be clearly recognizable. The cutoff frequency for letters was 1–2 cycles per letter width; faces were best recognized in a band centered at 4 cycles per face width. He also proposed that the perception of geometric visual illusions, such as the Mueller–Lyer and Poggendorf, was mediated by low spatial frequencies (Ginsberg, 1971, 1978; Ginsberg & Evans, 1979).

An issue that is related to the lowest frequency band that suffices for recognition is the encoding economy of a band. For a filter with a bandwidth that is proportional to frequency (e.g. a two-octave-wide filter), the lower the frequency, the smaller the number of frequency components needed to encode the filtered image of a constant object. Combining these two notions, Ginsburg concluded that objects were best, or most efficiently, characterized by the lowest band of spatial frequencies that sufficed to discriminate them. Ginsburg (1980) went on to suggest that higher spatial frequencies were redundant for certain tasks, such as face or letter recognition.

Several investigators were quick to point out that objects can be well discriminated in various spatial frequency bands. Fiorentini, Maffei and Sandini (1983) observed that faces were well recognized in either high or in lowpass filtered bands. Norman and Erlich (1987) observed that high spatial frequencies were essential for discrimination between toy tanks in photographs.

*To whom reprint requests should be addressed.

With respect to geometric illusions, both Janez (1984) and Carlson, Moeller and Anderson (1984) observed that the geometric illusions could be perceived for images that had been highpass filtered so that they contained no low spatial frequencies. This suggests that low and high spatial frequency bands may carry equivalently useful information for higher visual processes.

Characterizing the visual system

In the studies cited above, the discussion of spatial filtering focuses on *object* spatial frequencies, that is, frequencies that are defined in terms of some dimension of the object they describe (cycles per object). Most psychophysical research with spatial frequency bands has focused on *retinal* spatial frequencies, that is, frequencies defined in terms of retinal coordinates. For example, the spatial contrast sensitivity function (Davidson, 1968; Campbell & Robson, 1968) describes the threshold sensitivity of the visual system to sine wave gratings as a function of their *retinal* spatial frequency. Visual system sensitivity is greatest at 3–10 cycles per degree of visual angle (c/deg). How does visual system sensitivity relate to object spatial frequencies?

Unconfounding retinal and object spatial frequencies

Retinal spatial frequency and object spatial frequency can be varied independently to determine whether certain object frequencies are best perceived at particular retinal frequencies. Object frequency is manipulated by varying the frequency band of bandpass filtered images; retinal frequency is manipulated by varying the viewing distance.

The cutoff *object* spatial frequency of lowpass filters and the observer's viewing distance were varied independently by Legge, Pelli, Rubin and Schleske (1985) who studied reading rate of filtered text at viewing distances over a 133:1 range. Over about a 6:1 middle range of distances, reading rate was perfectly constant, and it was approximately constant over a 30:1 range. At the longest viewing distances, there was a sharp performance decrease (as the letters became indiscriminably small). At the shortest viewing distance, performance decreased slightly, perhaps due to large eye movements that the subjects would have to execute to bring relevant material towards their lines of sight, and to the impossibility of peripherally previewing new text.

While viewing distance changed the overall level of performance in Legge et al., the cutoff *object* frequency of their low-pass filters at which performance asymptoted did not change. From this study, we learn that reading rate can be quite independent of retinal frequency over a fairly wide range, and that dependence on critical object frequency does not depend on viewing distance. Because the authors measured reading rate only in lowpass filtered images, we cannot infer reading performance in higher spatial frequency bands from their data.

Unconfounding object statistics and visual system properties

Human visual performance is the result of the combined effects of the objectively available information in the stimulus, and the ability of humans to utilize the information. In studying visual performance with differently filtered images, it it critical to separate availability from ability to utilize. For example, narrow-band images can be completely described in terms of a small number of parameters—Fourier coefficients or any other independent descriptors—than wide-band images. Poor human performance with narrow-band images may reflect the impoverished image rather than an intrinsically human characteristic—an ideal observer would exhibit a similar loss.

The problem of assessing the utility of stimulus information becomes acute in comparing human performance in high and in low frequency bandpass filtered images. Typically, filters are constructed to have a bandwidth proportional to frequency (constant bandwidth in terms of octaves). For example, Ginsburg (1980) used faces filtered into 2-octave-wide bands; while Norman and Ehrlich (1987) also used 2-octave bands for their filtered tank pictures. With such filters, high spatial frequency images contain more independent frequencies than low frequency images.

Although linear bandwidth represents perhaps the important difference between images filtered in octave bands at different frequencies, the informational content of the various bands also depends critically on the nature of the specific class of objects, such as faces or letter. Obviously, determining the information content of images is a difficult problem. When it is not solved, the amount of stimulus information available within a frequency band is confounded

with the ability of human observers to use the information. Direct comparisons of performance between differently filtered objects are inappropriate. This distinction between objectively available stimulus information and the human ability to use it has not been adequately posed in the context of spatial bandpass filtering.

Efficiency

In the present context, physically available information is best characterized by the performance of an ideal observer. If there were no noise in the stimulus, the ideal observer would invariably respond perfectly. To compare the performance of an observer, human or ideal, noise of root-mean-square (r.m.s.) amplitude n is progressively added to the signal of r.m.s. amplitude s until the performance is reduced to some criterion, such as 50% correct in a letter identification task. This defines the signal to noise ratio, $(s/n)_c$, for a criterion c. Efficiency *eff* of human performance is defined by:

$$eff = \left(\frac{s_i}{n_i}\right)_c^2 \Big/ \left(\frac{s_h}{n_h}\right)_c^2$$

where h and i indicate *human* and *ideal* observers, and s and n are r.m.s. signal and noise amplitudes (Tanner & Birdsall, 1958). In a pure, quantally limited system, efficiency actually represents the fraction of quanta absorbed (utilization efficiency). In the context of signal detection theory, efficiency is given by a d' ratio:

$$eff = (d'_h/d'_i)^2.$$

Overview

For an object that contains a broad spectrum of spatial frequencies, object spatial frequency is determined by the center frequency of a spatial bandpass filtered image. Retinal spatial frequency is determined by the viewing distance at which the stimulus is viewed. Stimulus information is determined jointly by the signal-to-noise ratio, by the spatial filtering, and by the characteristics of the set of signals; these three informational components are combined in the efficiency computation. Letters are a convenient stimulus to study because they are highly overlearned so that human performance can be expected to be reasonably efficient, and because much is already known about the visibility of letters in the presence of internal noise (letter acuity) and about the visual processing of letters.

Specifically, to determine the roles of object and retinal spatial frequencies, letters are filtered into various frequency bands. Noise is added, and the psychometric function for correct identification is determined as a function of s/n. Accuracy depends only on s/n and not on overall contrast, for a wide range of contrasts (Pavel, Sperling, Riedl & Vanderbeck, 1987). This determination is repeated for every combination of object frequency band and viewing distance. Thereby, retinal spatial frequency and object spatial frequency are unconfounded, enabling us to determine whether a particular object frequency band is better discriminated in one visual channel (retinal frequency) than any other (Parish & Sperling, 1987a, b). Moreover, by computing an ideal observer for the identification task, we obtain an objective measure of the information that is present in each of the frequency bands. Finally, the comparison of human performance with the performance of the ideal observer gives us a precise measure of the ability of our subjects to utilize the information in the stimulus. Having untangled these factors, we can determine which spatial frequencies most efficiently characterize letters for identification.

METHOD

Two experiments were conducted using similar stimuli and procedures.

Stimuli

Letters (signals) and noise. The original, unfiltered letters were selected from a simple 5 × 7 upper-case font commonly used on CRT terminals. Since this is an experiment in pattern recognition, we felt that the simplest letter pattern might be the most general; indeed, this font has been widely used in letter discrimination studies. For the purpose of subsequent spatial filtering, the letters were redefined on a pixel grid that measured 45 (vertical height) × 35 (maximum horizontal extent of letters M and W). The letters had value 1 (white); the background had value 0 (black). To avoid edge effects in filtering, the background was extended to 128 × 128 pixels for all computations. However, only the center 90 × 90 pixels of the stimulus were displayed, as these contained effectively all the usable stimulus information, even for low spatial-frequency stimuli. Letters for presentation were chosen pseudo-randomly from the set of 26 upper-case English letters. Noise

Table 1. Parameters of the bandpass filters: lower and upper half-amplitude frequencies, peak, and 2D mean frequencies in cycles/letter height

Band	Lower	Peak	Upper	Mean[a]
0	0	Lowpass	0.53	0.39
1	0.26	0.53	1.05	0.74
2	0.53	1.05	2.11	1.49
3	1.05	2.11	4.22	2.92
4	2.11	4.22	8.44	5.77
5	6.33	Highpass	22.5	20.25

[a]Frequencies are weighted according to their squared amplitude (power) in computing the mean.

fields were defined on a 128 × 128 array by choosing independent Gaussian noise samples for each pixel, with the mean equal to zero and a variance σ^2 as required by the condition. (As with the letters, only the central 90 × 90 pixels were displayed.) Forty different noise fields were created.

Filters. Each stimulus consisted of a filtered letter added to an identically filtered noise field. Six spatial filters were available, corresponding to six successive levels of a Laplacian pyramid (Burt & Adelson, 1983). The zero-frequency component was added to the images so that they could be viewed. The object-relative filter characteristics, upper and lower half-amplitude cutoff and 2D mean frequency (cycles per letter height), appear in Table 1. The 2D mean frequency $\bar{f}$ for a given band is:

$$\bar{f} = \sum_{x=0}^{127}\sum_{y=0}^{127} f_{x,y}a_{x,y}^2 \Big/ \sum_{x=0}^{127}\sum_{y=0}^{127} a_{x,y}^2$$

where $f_{x,y}$ is the 2D frequency and $a_{x,y}$ is its amplitude. Cycles per object height is used rather than the more usual cycles per object width because the height of our upper-case letters remained constant across the entire set, whereas the width varied between letters.

The transfer functions (spectra) of the filters are displayed in Fig. 1. Approximately, filters are separated in spatial frequency by an octave (factor of 2) and have a bandwidth at half-amplitude of two octaves. The small mound in the lower right corner of Fig. 1 is a negligible imperfection in filter 4. For convenience, the limited range of spatial frequencies passed by each of the filters will be referred to as the *band* of that filter; a specific band is b_i ($i = 0, 1, 2, 3, 4, 5$), where b_0 is the lowest set of frequencies and b_5 is the highest.

The filter spectra (shown in Fig. 1) are approximately symmetrical in log frequency coordinates, a symmetrical spectrum in log coordinates is highly skewed to the right in linear frequency coordinates, resulting in a mean that

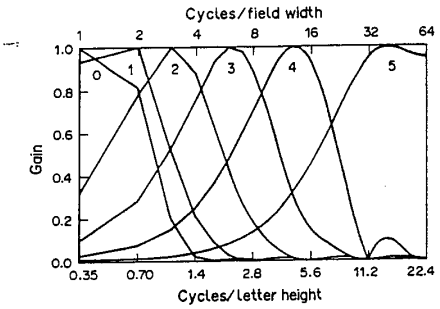


Fig. 1. Filter characteristics for the filters used in the experiments. There are two abscissas, both on a log scale. The top abscissa is the frequency in cycles per unwindowed field width (128 pixels); the bottom abscissa is in cycles per letter height (45 pixels). The ordinate is the normalized gain. The parameter i indicates the filter designation b_i in the text.

is much greater than the mode. In a 2D (vs 1D) filter, the rightward shift is accentuated. For example, band 2 has a peak frequency of 1.05 c/object but a 2D mean frequency of 1.49 c/object. The single most informative characterization of such a skewed bandpass spectrum depends somewhat on the context; usually use the mean rather than the peak.

Figure 2 (top) shows the letter G, filtered in bands 1–5 without noise; the bottom shows the same signals plus noise, $s/n = 0.5$. The full 128 × 128 array (extended by reflection beyond its edges) was passed through the filter so that the effect of the picture boundary did not intrude into the critical part of the display.

Signal to noise ratio, s/n. A filtered letter is a *signal.* Let i, j index a particular pixel in the x, y coordinate space of the stimulus. The signal contrast $c_s(i,j)$ of pixel i, j is:

$$c_s(i,j) = \frac{(l(i,j) - l_0)}{l_0} \qquad (1)$$

where $l_{i,j}$ is the luminance of pixel i, j and l_0 is the mean signal luminance over the 90 × 90 array. Signal power per pixel, s, is defined as mean contrast power averaged over the 90 × 90 pixel array:

$$s = (IJ)^{-1}\sum_{i}^{I}\sum_{j}^{J} c_s(i,j)^2 \qquad (2)$$

where $c_{i,j}$ is the contrast of pixel i, j and $I = J = 90$.

Noise contrast $c_n(i,j)$ is the value of the i,jth noise sample divided by the mean luminance. Analogously to signal power (equation 2), noise contrast power per pixel, n, is equal to $(\sigma/l_0)^2$. The signal to noise ratio is simply s/n.

Fig. 2. Top: unfiltered noise and unfiltered letter G. Middle: the letter G filtered in spatial frequency bands 1 5 with only quantization noise. Bottom: filtered letter G plus filter noise in the same bands with a signal-to-noise ratio of 0.50 in all panels. The effective s/n in the reproduction is somewhat lower (from Parish & Sperling, 1987a). The first row of numerals indicates the number by which the filter band is referred to in the text; the bottom row indicates the *mean* frequency of the bands in cycles per letter height.

Quantization. Our display system produced 256 discrete luminance levels. Level 128 was used as the mean luminance l_0; l_0 was 47.5 cd/m^2. To produce a visual display of a given letter, band, and s/n, signal power s and noise power n were normalized so that the luminance of every one of the 8100 displayed pixels fell within the range of the display system; there was no truncation of the tails of the Gaussian noise. (Although the relationship between input gray-level and output luminance was not quite linear at the extreme intensity values, it was determined that more than 90% of the pixels fell within the linear intensity range.) Intensity normalization was applied separately to each stimulus (combination of signal plus noise). By normalizing the total stimulus $s+n$, the actual value of s displayed to the subject diminished as n increased; i.e. the actual value of s was not known by the subject. Indeed, even stimuli with precisely the same letter in the same band and with the same s/n might be produced with slightly different s and n depending on the extreme values of the noise fields.

Seven values of s/n were available for each band, chosen in a pilot study to insure that the data yielded the entire psychometric function (chance to best performance). The same pilot study showed that subjects never performed above chance when confronted with noise-free letters from b_0; this band was omitted from the present study.

Procedure: experiment 1

Four of the experimental variables—letter identity, noise field, frequency band, and s/n—were randomized within each session. A fifth variable, viewing distance, was held constant within each session and was varied between sessions. Four viewing distances were used: 0.121, 0.38, 1.21 and 3.84 m. A chin rest was used to stabilize the subject's head for viewing at the shortest distance. At the four distances, the 90 × 90 pixel stimulus subtended 31.6, 10, 3.16 and 1.0 deg of visual angle respectively. The upper and lower half-amplitude cut-off retinal frequencies for the upper six filters, with respect to the four viewing distances used in this experiment, and for a fifth distance used in the second experiment, appear in Table 2. Subjects participated in four 1-hr sessions at each viewing distance. Each session consisted of 315 trials, nine trials at each of seven s/n's for each of the five frequency bands.

Prior to the first session, subjects were shown noise-free examples of the unfiltered letters. They were told that each stimulus presentation consisted of a letter and a certain amount of noise, and that the letter may appear degraded in some way. They were informed that at no time would a letter be shifted in orientation or from its central location in the stimulus field. Finally, they were instructed to view each stimulus for as long as they desired before making their best guess as to which letter had been presented. A response (letter identity) was required on every trial. Subjects typed the response on a keyboard connected to the host computer (Vax 11/750); subsequently, typing a carriage return erased the video screen and initiated the next trial in a few seconds. The room illumination was very dim: the response keyboard was lighted by stray light from its associated CRT terminal. No feedback was offered to the subjects.

Observers

Three subjects, two male and one female, between the ages of 20 and 27 participated in the experiment. All subjects had normal or corrected-to-normal vision. One of the subjects was a paid participant in the study.

Procedure: experiment 2

This experiment was run before expt 1. It is reported here because it offers additional data with two new and one old subject at a fifth viewing distance. Except as noted, the procedures are similar to expt 1. The screen was viewed through a darkened hood at a distance

Table 2. Lower and upper half-power frequency and 2D mean frequency (in c deg of visual angle) for all bands and viewing distances used in both experiments

Band	Viewing distance (m) 0.12	0.38	1.21	3.84	0.48
0 (lowpass)	0.00–0.04 (0.03)	0.00–0.12 (0.09)	0.00–0.37 (0.27)	0.00–1.18 (0.87)	0.00–0.15 (0.11)
1	0.02–0.07 (0.05)	0.06–0.23 (0.16)	0.18–0.74 (0.52)	0.58–2.34 (1.65)	0.07–0.29 (0.21)
2	0.04–0.15 (0.10)	0.12–0.47 (0.33)	0.37–1.48 (1.04)	1.18–4.70 (3.30)	0.15–0.59 (0.41)
3	0.07–0.30 (0.20)	0.23–0.94 (0.64)	0.74–2.97 (2.04)	2.34–9.40 (6.48)	0.29–1.18 (0.81)
4	0.15–0.59 (0.40)	0.47–1.88 (1.27)	1.48–5.94 (4.04)	4.70–18.80 (12.82)	0.59–2.36 (1.60)
5 (highpass)	0.30–2.25 (1.41)	0.94–7.13 (4.45)	2.97–22.53 (14.19)	9.40–71.27 (45.00)	1.77–8.96 (5.63)

e., letters are huge. The 7 point character subtends 0.33 deg at 0.3 m

of 0.48 m. At this distance, the 90 × 90 stimuli subtended 7.15 deg of visual angle. The half-amplitude cut-off frequencies and the mean frequencies of the six spatial filters are given in the rightmost column of Table 2. Three male subjects between the ages of 20 and 27 participated in the experiment. All subjects had normal or corrected-to-normal vision. Two of the subjects were paid for their participation, and one, DHP, also participated in expt 1. Five sessions of 315 trials were run for each subject.

RESULTS

Psychometric functions: $\hat{p}$ vs $\log_{10} s/n$

The measure of performance is the observed probability $\hat{p}$ of a correct letter identification. The complete psychometric functions are displayed in Figs 3 (expt 1) and 4 (expt 2). A separate psychometric function is shown for each subject, viewing distance and frequency band. In band b_1, for all subjects, performance asymptotes (for noiseless stimuli) at $\hat{p} \approx 0.5$. In all other bands, performance improves from near-chance (1/26) to near perfect as the value of s/n increases.

Noise resistance as a function of frequency band

An obvious aspect of the data of both experiments is that the data move to the left of the figure panels as band spatial frequency increases. This means that high spatial frequency stimuli (bands b_4, b_5) are identifiable at smaller

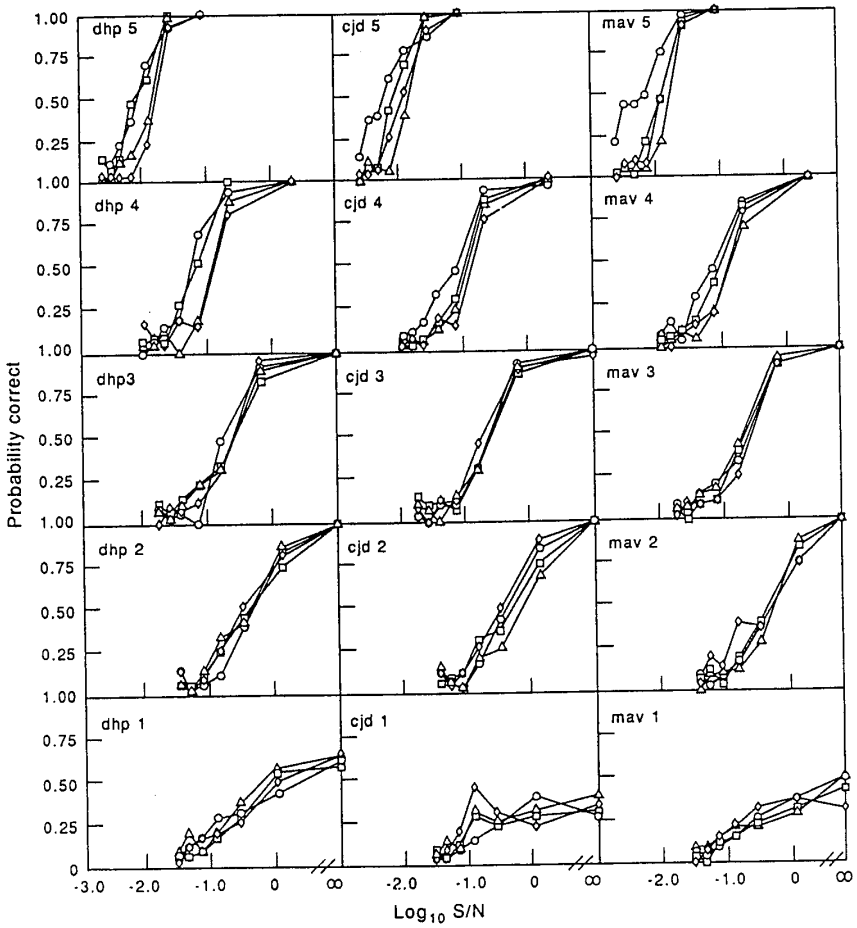


Fig. 3. Psychometric functions from expt 1. Each graph displays performance as a function of $\log_{10} s/n$, within a frequency band. The parameter is viewing distance. Subjects are arranged in columns and frequency band is arranged in rows, progressing from the highest frequency band at the top to the lowest band at the bottom. The four viewing distances are 3.84 (○), 1.21 (△), 0.38 (□), and 0.121 (◇) m.

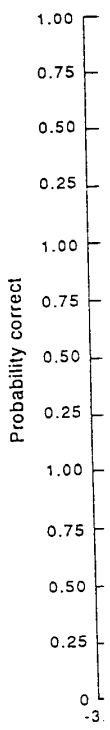


Fig. 4. Psyc
quency banc
five frequenc
○, □, △, ◇
is

s/n than s
noise incr
enable cc
function c
estimated
from exp
from the
ing distar
made usi
distance
$(s/n)_{50\%}$ f
frequency
For comp
ment in (
ber of fre
low to h
series of
improves
than 1 f
results, a
the Disc

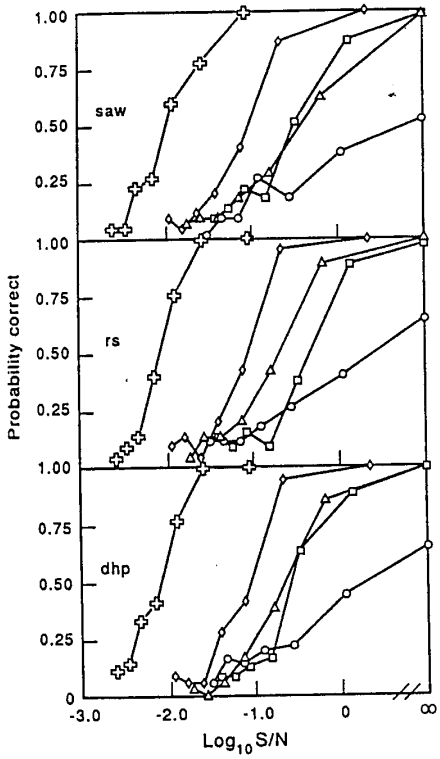


Fig. 4. Psychometric functions for each subject and frequency band in expt 2. Viewing distance was 0.48 m. The five frequency bands, b_1–b_5, are indicated, respectively, by ○, □, △, ◇ and +. The probability of a correct response is plotted as a function of $\log_{10} s/n$.

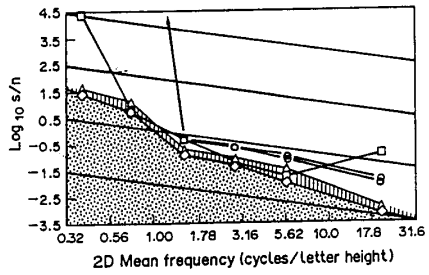


Fig. 5. Performance of human subjects and various computational discriminators. The abscissa indicates $\log_{10}$ of the mean frequency of each bandpass stimulus. The ordinate indicates the (interpolated) s/n ratio at which a probability of a correct response $p = 0.5$ is achieved. Circles indicate each of the three subjects in expt 1 at the intermediate viewing distance of 1.21 m. In band b_1, 2 of 3 human subjects fail to achieve 50% correct ($eff = 0$); these points lie outside the graph. (△) indicates sub-ideal and (◇) indicates super-ideal performances of discriminators that brackets the ideal discriminator. The shaded area below the super-ideal discriminator indicates theoretically unachievable performance. Squares indicate performance of a spatial correlator-discriminator. The oblique parallel lines have slope −1 that represents the improvement in expected performance (decrease in s/n) as function of the number of frequency components in each band when filter bandwidth is proportional to frequency.

s/n than stimuli in bands b_1 and b_2; resistance to noise increases with spatial frequency band. To enable comparisons of noise sensitivity as a function of band, the s/n at which $\hat{p} = 50\%$ was estimated for each subject and frequency band from expt 1 by means of inverse interpolation from the best fitting logistic function. As viewing distance had no effect, all estimates were made using the data collected when viewing distance was equal to 0.38 m. A graph of these $(s/n)_{50\%}$ points as a function of the mean object frequency of the band is plotted in Fig. 5 (○). For comparison, the expected rate of improvement in $(s/n)_{50\%}$, based on the increasing number of frequency components as one moves from low to high frequency bands, is plotted as a series of parallel lines in Fig. 5. Performance improves [$(s/n)_{50\%}$ decreases] somewhat faster than $1/f$ (the slope of the parallel lines). These results, and Fig. 5, will be analyzed in detail in the Discussion section.

The non-effect of viewing distance

Another property of the data is that, in most conditions, viewing distance has no effect on performance. Analysis of variance, carried out individually for each subject, shows that there is no significant effect of distance in any band for subject dhp and a significant effect of distance in bands b_4 and b_5 for the other two subjects. Further analysis by a Tukey test (Winer, 1971) in bands b_4 and b_5 for these subjects shows that the only significant effect of distance is that visibility at the longest viewing distance is *better* than at the other three distances. For subject CJD, the improvement is equivalent to a gain in s/n of 0.19 and 0.28 $\log_{10}$ (for bands b_4 and b_5, respectively); for MAV, the corresponding gains were 0.21 and 0.40.

Improved performance at long viewing distances is almost certainly due to the square configuration of individual pixels, which produces a high frequency spatial pixel noise that is attenuated by viewing from sufficiently far away (Harmon & Julesz, 1973). In low frequency bands, pixel-boundary noise is not a problem because the spatial filtering insures that adjacent pixels vary only slightly in intensity. We explored the hypothesis of pixel-boundary noise with subject CJD, who showed a distance effect

in band 5. At an intermediate viewing distance of 1.21 m, CJD squinted her eyes while viewing stimuli from band 5. By blurring the retinal image of the display in this way, performance improved approximately to the level of the furthest viewing distance.

To summarize, the only significant effect of distance that we observed was a lowering of performance at near viewing distances relative to the furthest distance. This impairment occurred primarily in bands 4 and 5. In these bands, the spatial quantization of the display (90 × 90 square-shaped pixels) produces artifactual high spatial frequencies that mask the target. These artifactually produced spatial frequencies can be attenuated by deliberate blurring (squinting), or by producing displays with higher spatial resolution, or by increasing the viewing distance to the point where the pixel boundaries are attenuated by the optics of the eye and neural components of the visual modulation transfer function. In all cases, blurring improves performance and eliminates the slightly deleterious effect of a too small viewing distance. Thus, for correctly constructed stimuli, in the frequency ranges studied, there would be no significant effect of viewing distance on performance. This finding is in agreement with the results of Legge et al. (1985), who examined reading rate rather than letter recognition. It is in stark disagreement with the results of sinewave detection experiments in which retinal frequency is critical—see Sperling (1989) for an explanation.

DISCUSSION

A comparison of performance in different frequency bands shows that subjects perform better the higher the frequency band; and subjects require the smallest signal-to-noise ratio in the highest frequency band. To determine whether performance in high frequency bands is good because humans are more efficient in utilizing high-frequency information, or because there is objectively more information in the high-frequency images, or both, requires an investigation of the performance of an ideal observer. The performance of the ideal observer is the measure of the objective presence of information. Human performance results from the joint effect of the objective presence of information and the ability of humans to utilize that information. Human efficiency is the ratio of human performance to ideal performance.

Ideal discriminator

Definition. An ideal discriminator makes the best possible decision given the available data and the interpretation of "best." The performance of the ideal discriminator defines the objective utility of the information in the stimulus. We prefer the name *ideal discriminator*, rather than *ideal observer*, because it indicates the critical aspect of performance under consideration, but we occasionally use *ideal observer* to emphasize the relations to a large, relevant literature on this subject. Our purposes in this section are first, to derive an ideal discriminator for the letter identification task, second, to develop a practical working approximation to this discriminator, and third, to compare the performance of the human with the ideal discriminator.

Although ideal observers have recently come into greater use in vision research, the applications have focused primarily on determining the limits of performance for relatively low-level visual phenomena. For example, Barlow (1978, 1980), and Barlow and Reeves (1979) investigated the perception of density and of mirror symmetry; Geisler (1984) investigated the limits of acuity and hyperacuity; Legge, Kersten and Burgess (1987) examined the pedestal effect; Kersten (1984) studied the detection of noise patterns; and Pelli (1981) detailed the roles of internal visual noise. Geisler (1989) provides an overview of efficiency computations in early vision. Our application differs from these in that we expand the techniques and apply them to a higher perceptual/cognitive function, letter recognition.

For the letter identification task, the ideal discriminator is conceptually easy to define. A particular observed stimulus, x, representing an unknown letter plus noise, consists of an intensity value (one of 256 possible values) at each of 90 × 90 locations. The discriminator's task is to make the correct choice as frequently as possible from among the 26 alternative letters.

The likelihood of observing stimulus x, given each of the 26 possible signal alternatives, can be computed when the probability density function of the added noise is known exactly. The optimal decision chooses the letter that has the highest likelihood of yielding x. The expected performance of the ideal discriminator is computed by summing its probability of a correct response over the 256^{8100} possible stimuli (256 gray levels, 90 × 90 pixels). Unfortunately,

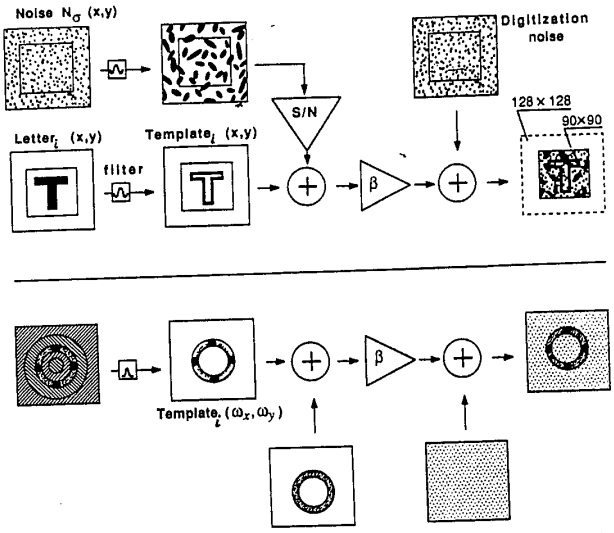


Fig. 6. Flow chart of the experimental procedures that are modelled by the ideal discriminator analysis. Upper half indicates space-domain operations; lower half indicates the corresponding operations in the frequency domain. Computations are carried out on 128 × 128 arrays; the subject sees only the center 90 × 90 pixels. A random letter and a random noise field are each filtered by the same filter (b); the noise is amplified to provide the desired signal-to-noise ratio; the letter and noise are added, the output is scaled and quantized (represented by the addition of digitization noise), and the result is shown to the subject. In the frequency domain ω_x, ω_y, the bandpass filter selects an annulus, whereas the quantization noise is uniform over ω_x, ω_y.

when there is both bandpass filtered and intensity quantization, the usual simplifications that make this enormous computation tractable are not applicable.

As an alternative to computing the expected performance of the ideal discriminator, one can compute its performance with a particular subset of the possible stimuli—the stimuli that the subject actually viewed or, preferably, a larger set of stimuli for more reliable estimation. This Monte Carlo simulation of the performance of the ideal discriminator is a tractable computation that yields an estimate of expected performance.

Derivation. Stimulus construction is diagrammed in Fig. 6 which shows the equivalent operations in the space and the frequency domains. To derive an ideal discriminator, we need to carefully review the processes of stimulus construction. We use uppercase letters to represent quantities in the frequency domain and lowercase letters to represent quantities in the space domain. A letter is defined by a 90 × 90 array that takes the value 1 at the letter locations and 0 at the background locations. When this array is spatially filtered in band b, it defines the *letter template* $t_{i,b}(x, y)$, where i indicates the particular letter, b the frequency band, and x, y the pixel location. We write $T_{i,b}(\omega_x, \omega_y)$ for the Fourier series coefficient of $t_{i,b}$ indexed by frequency.

An unknown stimulus $u_{i,b}(x, y)$ to be viewed by a subject is produced by adding filtered $n_b(x, y)$ with post-filtering variance σ_N^2, to the template $t_{i,b}(x, y)$, where letter identity i is unknown to the subject. The stimulus is scaled and digitized (quantized) to 256 levels prior to presentation, contributing an additional source of noise $q_{i,b}(x, y)$, called digitization noise. Finally, a d.c. component (dc) is added to $u_{i,b}$ to bring the mean luminance level to 128. These steps are diagrammed in Fig. 6 which shows both the space-domain and the corresponding frequency-domain operations. The space-domain computation is encapsulated in equations (3):

$$u_{i,b}(x, y) = \beta_{i,b}[t_{i,b}(x, y) + n_b(x, y)] \quad (3a)$$

$$u_{i,b}(x, y) = \beta_{i,b}[t_{i,b}(x, y) + n_b(x, y)] + q_{i,b}(x, y) + dc. \quad (3b)$$

The scaling constant $\beta_{i,b}$, limits the range of real values for each pixel, prior to quantization, to $[-0.5, 255.5]$. The degree of scaling is determined by the maximum and minimum values in

the function $t_{i,b} + n_b$. Note that the extreme values in the image are determined by σ_{N_2} which is adjusted to yield the appropriate s/n for each condition; the values of $t_{i,b}$ are fixed prior to scaling. Specifically:

$$\beta_{i,b} = \frac{256}{\max(t_{i,b} + n_b) - \min(t_{i,b} + n_b)}. \quad (4)$$

As a result of bandpass filtering, the noise samples in adjacent pixels are strongly dependent on each other. Therefore, the discriminator problem is best approached in the Fourier domain, where the random variables $\{N_b(\omega_x, \omega_y)\}$ are jointly independent because the filtering operations simply scale the different frequency components without introducing any correlations (van Tress, 1968). The task of the ideal discriminator is to pick the template $t_{i,b}$ that maximizes the likelihood of $u_{i,b}$ with *a priori* knowledge of: (i) the fixed functions $t_{i,b}$, and their probabilities; and (ii) the densities of the jointly independent random variables $\{N_b(\omega_x, \omega_y)\}$. As is clear, $\beta_{i,b}$, σ_N^2, $\{Q_{i,b}(\omega_x, \omega_y)\}$, and $\{N_{i,b}(\omega_x, \omega_y)\}$ are all jointly distributed random variables characterized by some density f. To compute the likelihood of $u_{i,b}$ the ideal discriminator must integrate f over all possible values that may be assumed by the set of jointly, distributed random variables, whose values are constrained only in that they result in a possible stimulus $u_{i,b}$. Unfortunately, no closed-form solution to this problem is available, forcing us to look for an alternative approach.

Bracketing. To estimate the performance of the ideal discriminator, we look for a tractable super-ideal discriminator that is better than the ideal but which is solvable. Similarly, we look for a tractable sub-ideal discriminator that is worse than the ideal. The ideal discriminator must lie between these two discriminators; that is, we bracket its performance between that of a "super-ideal" and a "sub-ideal" discriminator. The more similar the performance of the super- and sub-ideal discriminators, the more constrained is the ideal performance which lies between them.

Our super-ideal discriminator is told, *a priori*, the extact values for $\beta_{i,b}$ and σ_N^2 for each stimulus presentation. Therefore, it is expected to perform slightly better than the ideal discriminator which must estimate these values from the data. The sub-ideal discriminator estimates these same parameters from the presented stimulus in a simple but nonideal way. Therefore, it is expected to perform slightly worse than the ideal discriminator. The computational forms used to compute $\beta_{i,b}$ and σ_N^2 for the sub-ideal discriminator are presented in the Appendix, along with the derivation of the likelihood estimator used by both discriminators. A complete discussion of these derivations and the problems associated with the formulation of an ideal discriminator for such complex stimuli is presented in Chubb, Sperling and Parish (1987).

Performance of the bracketed discriminator. The super- and sub-ideal discriminators were tested in a Monte Carlo series of trials, in which they each were confronted with 90 stimuli in each of the frequency bands at each of seven s/n values chosen to best estimate their 50% performance point. The s/n necessary for 50% correct discriminations was estimated by an inverse interpolation of the best fitting logistic function. The derived $(s/n)_{50\%}$ is the measure of performance of a discriminator. The mean ratio, across frequency bands, of

$$(s/n)_{50\%} \text{ sub-ideal}/(s/n)_{50\%} \text{ super-ideal}$$

is about 2 (approx. 0.3 $\log_{10}$ units). The ratio does not depend on the criterion of performance.

Efficiency of human discrimination

In all conditions, human subjects perform worse than the sub-ideal discriminator. Notably, with no added luminance noise, the subideal (and, of course, the ideal) discriminator function perfectly, even in b_0 where subject performance is at chance, and in b_1 where subjects reached asymptote at about 50% correct.

Data from the subjects are plotted with the $(s/n)_{50\%}$ sub-ideal and $(s/n)_{50\%}$ super-ideal in Fig. 5. For comparison, Fig. 5 also shows the performance of a correlator discriminator which chooses the letter template that correlates most highly with the stimulus in the space domain. In the coordinates of Fig. 5 ($\log_{10} s/n$ vs $\log_{10} f$ where f represents the mean 2D spatial frequency of the band), the vertical distance d from the human data $\log(s/n)_{50\%}$, *human* down to the bracketed discriminator $\log(s/n)_{50\%}$, *ideal* represents the $\log_{10}$ of the factor by which the bracketed discriminator outperforms the human observer at that value of f. For the purpose of specifying efficiency, we assume the ideal discriminator lies at the mid-point of the sub and super-ideal discriminators in Fig. 5. The

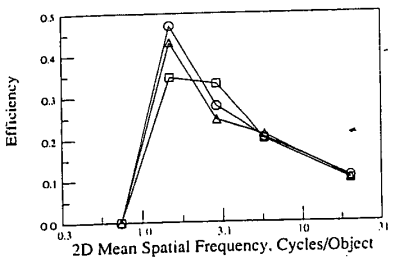


Fig. 7. Discrimination efficiency as a function of the mean frequency of a 2-octave band (in cycles per letter height) indicated on a logarithmic scale. Data are shown for three observers: △ = SAW, □ = RS, ○ = DHP. The viewing distance is 2.21 m, which is representative of all viewing distances tested.

efficiency *eff* of human discrimination relative to the bracketed discriminator is $eff = 10^{-2d}$, where:

$$d = \log(s/n)_{50\%,\,human} - \log(s/n)_{50\%,\,ideal}.$$

The values of *eff* in each object frequency band are shown in Fig. 7. In band 0, *eff* is zero because human performance never reaches 50%; indeed, it never rises significantly above 4% (chance). In band 1, human performance asymptotically climbs close to 50% as s/n approaches infinity; $eff \approx 0$. In band 2, *eff* reaches its maximum of 35–47% (depending on the subject), and it declines rapidly with increasing frequency (b_3–b_5).

The 42% average efficiency in band 2 is similar in magnitude to the highest efficiencies observed in comparable studies. For example, efficiency has been determined for detecting various kinds of patterns in arrays of random dots (Barlow, 1978, 1980; van Meeteren & Barlow, 1981), tasks which, like ours, may require significantly cognitive processing. In a wide range of conditions, the highest efficiencies observed were about 50%, and frequently lower. Van Meeteren and Barlow (1981) also found that efficiency was perfectly correlated with object spatial frequency and was independent of retinal spatial frequency.

Spatial correlator discriminator. A correlator discriminator cross-correlates the presented stimulus with its memory templates and chooses the template with the highest correlation. Correlation can be carried out in the space or in the frequency domain. Correlation is an efficient strategy when noise in adjacent pixels is independent and when members of the set of signals have the same energy; both of these conditions are violated by our stimuli. However, when sufficient prior information is available to subjects, they do appear to employ a cross-correlation strategy (Burgess, 1985).

It is interesting to note that the performance of the spatial correlator discriminator over the middle range of spatial frequencies is quite close to the performance of the sub-ideal discriminator. At high spatial frequencies, correlator performance degenerates, due to its inability to focus spatially on those pixel locations that contain the most information. A spatial correlator that optimally weighted spatial locations, could overcome the spatial focusing problem at high frequencies. (Spatial focusing is treated in the next section.)

At all frequencies, the spatial correlator is nonideal because noise at spatial adjacent pixels is not independent. At low spatial frequencies, the nonindependence of adjacent locations becomes extreme and the correlator fails miserably. This points out that, for our stimuli, correlation detection is better carried out in the frequency domain because there the noise at different frequencies is independent. The qualitative similarity between the correlator discriminator and the subjects' data suggests that the subjects might be employing a spatial correlation strategy, augmented by location weighting at high frequencies.

Lowest spatial frequencies sufficient for letter discrimination. Band 2 corresponds to a 2-octave band with a peak frequency of 1.05 c/object (vertical height of letters) and a 2D mean frequency of 1.49 c/object. At the four viewing distances, 1.05 c/object corresponds to retinal frequencies of 0.074, 0.234, 0.739 and 2.34 c/deg of visual angle. We observe perfect scale invariance: all of these retinal frequencies, and hence the visual channels that process this information, are equally effective in achieving the high efficiency of discrimination.

The finding that b_2 with a center frequency of 1.05 c/object and a $\frac{1}{2}$ amplitude cutoff at 2.1 c/object is critical for letter discrimination is in good agreement with previous findings of both Ginsburg (1978) for letter recognition and Legge et al. (1985) for reading rate. Legge et al. used low-pass filtered stimuli, which included not only spatial frequencies within an octave of 1 c/object (b_2) but also included all lower frequencies. From the present study, we expect human performance with low-pass and with band-pass spatial filtering to be quite similar up to 1 c/object because the lowest frequency

bands, when presented in isolation, are perceptually useless (at least when presented alone).

It is an important fact that our subjects actually performed better, in the sense of achieving criterion performance at a lower s/n ratio, at higher frequency bands than b_2. This is explained by the increase in stimulus information in higher frequency stimuli. Increased information more than compensates for the subjects' loss in efficiency as spatial frequency increases.

Components of discrimination performance

Though the performance of the bracketed ideal discriminator is useful in quantifying the informational utility of the various bands, it is instructive to consider the changing physical structure of the stimuli as well. What components of the stimuli actually lead to a gain in information with increasing frequency? According to Shannon's theorem (Shannon & Weaver, 1949), an absolutely bandlimited 1-D signal can be represented by a number of samples m that is proportional to its bandwidth. When the signal-to-noise ratio in each sample s_i/n_i is the same, the overall signal-to-noise ratio s/n grows as $\sqrt{m}$. In the space domain, our filters were constructed (approximately) to differ only in scale but not in the shape of their impulse responses. Therefore, when the mean frequency of a filter band increased by a factor of 2, the bandwidth also increased by 2. Since the stimuli are 2D, the effective number of samples increases with the square of frequency, and the increase in effective s/n ratio is proportional to m. This expected improvement with frequency, based simply on the increase in effective number of samples, is indicated by the oblique parallel lines of Fig. 5 with slope of -1. The expected improvement in threshold s/n due simply to the linearly increasing bandwidth of the bands does a reasonable job of accounting for the improvement in performance for both human and bracketed discriminators between b_2 and b_5.

Performance of all discriminators improves faster with frequency between 0.39 and 1.5 c/object and between 5.8 and 22 c/object than is predicted from the bandwidths of the images. A slope steeper than -1 means that there is more information for discriminating letters in higher frequency bands even when the number of independent samples is kept the same in each band. Once sampling density is controlled, just how much information letters happen to contain in each frequency band is an ecological property of upper-case letters.

Increasing spatial localization with increasing frequency band. From the human observer's point of view, the letter information in low-pass filtered images is spread out over a large portion of the total image array. In high spatial-frequency images, the letter information is concentrated in a small proportion of the total number of pixels. In high spatial-frequency images, a human observer who knows which pixels to attend will experience an effective s/n that is higher than an observer who attends equally to all pixels. In this respect, humans differ from an ideal discriminator. The ideal discriminator has unlimited memory and processing resources, does not explicitly incorporate any selective mechanism into its decision, and uses the same algorithm in all frequency bands. Information from irrelevant pixels is enmeshed in the computation but cancels out perfectly in the letter-decision process. To understand human performance, however, it is useful to examine how, with our size-scaled spatial filters, letter information comes to be occupy a smaller and smaller fraction of the image array as spatial frequency increases.

Here we consider three formulations of the change in the internal structure of the images with increasing spatial frequency: (1) spatial localization; (2) correlation between signals; and (3) nearest neighbor analysis. We have already noted that, in our images, the information-rich pixels become a smaller fraction of the total pixels as frequency band increases. Indeed, this reduction can be estimated by computing the information transmitted at any particular pixel location or, more appropriately for estimating noise resistance, by computing the variance of intensity (at that pixel location) over the set of 26 alternative signals.

To demonstrate the degree of increasing localization with increasing frequency, the variance (over the set of 26 letter templates) was computed at each pixel location (x, y). *Total power*, the total variance, is obtained by summing over pixel locations. The number of pixel locations needed to achieve a specific fraction of the total power is given in Fig. 8, with frequency band as a parameter. These curves describe the spatial distribution of information in the latter templates. If all pixels were equally informative, exactly half of the total number of pixels would be needed to account for 50% of the total power. The solid curves in Fig. 8 show that the number of pixels needed to convey any percentage of total signal power, decreases as the

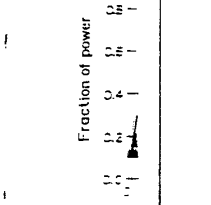


Fig. 8. Fra
extreme-va
lines indica
parameter
power frac
fractions fr
label, they a

frequency
distributi
our set o
be needed
The da
random
quency ba
power is
enormous
signal po
informati
spatial loc
Correla
way of d
with banc
less confu
frequency
ity is the
the 26 let
(Table 3)
letter tem
to 0.31 in
have a pa
whelming
mation").
wonder t
band. Ov
criminato
correlatio
way to u
templates
Nearest
neighbors
accuracy
of errors.
letter i as
8100 (90 >

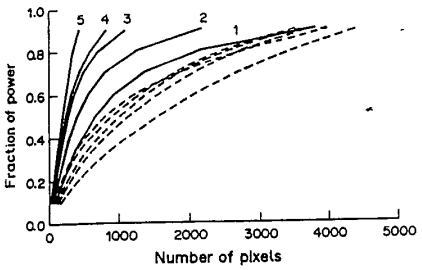


Fig. 8. Fraction of total power contained in the n most extreme-valued pixels as a function of n (out of 8100). Solid lines indicate the power fractions for signals; the curve parameter indicates the filter band. Dashed lines indicate power fractions for filtered noise fields. Although power fractions from successive bands of noise are too close to label, they generally fall in the same left–right 5–0 order as those for signal bands.

frequency band increases. These information distribution curves are an ecological property of our set of letter stimuli; different curves would be needed describe other stimulus sets.

The dashed curves in Fig. 8 were derived from random noise filtered in each of the six frequency bands (b_0–b_5). The distribution of noise power is very similar between the various bands, enormously more so than the distribution of signal power. For our letter stimuli, stimulus information coalesces to a smaller number of spatial locations as spatial frequency increases.

Correlation between signals. A more abstract way of describing the change of information with bandwidth is to note that letters become less confusible with each other in the higher frequency bands. A good measure of confusibility is the average pairwise correlation between the 26 letter templates in each frequency band (Table 3). The average correlation between letter templates diminishes from 0.94 in band 0 to 0.31 in band 5. In a band in which templates have a pairwise correlation over 0.9, the overwhelming amount of intensity variation ("information") is useless for discrimination. Small wonder that subjects fail completely in this band. Overall, performance of the ideal discriminator and of observers improves as the correlation decreases, but there is no obvious way to use the pairwise correlation between templates to predict performance.

Nearest neighbors. The analysis of nearest neighbors is a useful technique for predicting accuracy by the analysis of the possible causes of errors. We can regard a filtered image t_i of letter i as a vector in a space of dimensionality 8100 (90 × 90 pixels). When noise is added, the possible positions of t_i are described by a cloud whose dimensions are determined by the s/n ratio. A neighboring letter k may be confused with letter i when the cloud around t_i envelopes t_k. The closer the neighbor, the greater the opportunity for error. Table 3 gives the average normalized distance to the nearest neighbor in each of the bands. The increase in distance to the nearest neighbor reflects the improvement in the representation of signals as spatial frequency increases.

Table 3. Average pairwise correlations and nearest neighbors (Euclidean distance × 10^{-5})

Band	Correlations	Nearest neighbor
0	0.94	0.01
1	0.91	0.30
2	0.58	1.2
3	0.38	2.3
4	0.33	3.1
5	0.31	4.1

We consider possible causes of lower efficiency of discrimination in bands below b_2. The letters in these bands have high pair-wise correlations and the mean band frequency is less than the object frequency. This means that letters differ only in subtle differences of shading, a feature that we usually do not think of as shape. Observers would need to be able to utilize small intensity differences to distinguish between letters. To eliminate an alternative explanation (the smaller number of frequency components in the low-frequency bands), we conducted an informal experiment with a lower fundamental frequency. The fundamental frequency, which is outside the band, nevertheless determines the spacing of frequency components within the band. Reducing the fundamental frequency of the letter by one-half increases the number of frequency components in the band by a factor of 4. (A 256 × 256 sampling grid was used rather than 128 × 128.) These 4× more highly sampled stimuli were not more discriminable than the original stimuli. This suggests that the internal letter representation (template) that subjects bring with them to the experiment cannot utilize low-frequency information, even when it is abundantly available. Whether, with sufficient training, subjects could learn to use low spatial frequencies to make letter discriminations is an open question.

SUMMARY AND CONCLUSIONS

1. Visual discrimination of letters in noise, spatially filtered in 2-octave wide bands, is

independent of viewing distance (retinal frequency) but improves as spatial frequency increases.

2. The improvement in performance with increasing spatial frequency results mainly from an increase in the objective amount of information transmitted by the filters with increasing frequency (because filter bandwidth was proportional to center frequency) which is manifested as objectively less confusible stimuli in the higher bands.

3. The comparison of human performance with that of an estimated ideal discriminator demonstrates that humans achieve optimal discrimination (a remarkable 42% efficiency) when letters are defined by a 2-octave band of spatial frequencies centered at 1 cycle per letter height (mean frequency 1.5 c/letter). This high efficiency of discrimination is maintained over a 32:1 range of viewing distances.

4. Detection efficiency was invariant over a range of retinal spatial frequencies in which the contrast threshold for detection of sine gratings (the modulation transfer function, MTF) varies enormously. The independence of detection performance and retinal size held for all frequency bands.

5. A part of the loss of human efficiency in discrimination as spatial frequency exceeded 1 c/object height may have been due to the subjects' inability to identify, to selectively attend, and to utilize the smaller fraction of information-rich pixels in the higher frequency images.

6. Finally, it is important to note that without the comparison to the ideal observer, we would not have been able to understand the components of human performance in the different frequency bands.

Acknowledgements—We acknowledge the large contribution of Charles Chubb to the formulation and solution of the ideal discriminator. We thank Michael S. Landy for helpful comments and Robert Picardi for skillful technical assistance. The project was supported by USAF, Life Sciences Directorate, Visual Information Processing Program, grants 85-0364 and 88-0140.

REFERENCES

Barlow, H. B. (1978). The efficiency of detecting changes of density in random dot patterns. *Vision Research, 18*, 637–650.

Barlow, H. M. (1980). The absolute efficiency of perceptual decisions. *Philosophical Transactions of the Royal Society, London B, 290*, 71–82.

Barlow, H. B. & Reeves, B. C. (1979). The versatility and absolute efficiency of detecting mirror symmetry in random dot displays. *Vision Research, 19*, 783–793.

Burgess, A. (1985). Visual signal detection—III. On Bayesian use of prior knowledge and cross correlation *Journal of the Optical Society of America A, 2(9)*, 1498–1507.

Burgess, A. (1986). Induced internal noise in visual decision tasks. *Journal of the Optical Society of America A, 3*, 93

Burt, P. J. & Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications, Com-34(4)*, 532–540.

Campbell, F. W. & Robson, J. G. (1968). Application of Fourier analysis to the visibility of gratings. *Journal of Physiology, London 197*, 551–566.

Carlson, C. R., Moeller, J. R. & Anderson, C. H. (1984) Visual illusions without low spatial frequencies. *Vision Research, 24*, 1407–1413.

Chubb, C., Sperling, G. & Parish, D. H. (1987). Designing psychophysical discrimination tasks for which ideal performance is computationally tractable. Unpublished manuscript, New York University, Human Information Processing Laboratory.

Davidson, M. L. (1968). Perturbation approach to spatial brightness interaction in human vision. *Journal of the Optical Society of America A, 58*, 1300–1309.

Fiorentini, A., Maffei, L. & Sandini, G. (1983). The role of high spatial frequencies in face perception. *Perception, 12*, 195–201.

Geisler, W. S. (1984). Physical limits of acuity and hyperacuity. *Journal of the Optical Society of America A, 1*, 775–782.

Geisler, W. S. (1989). Sequential ideal-observer analysis of visual discriminations. *Psychological Review, 21*, 267–314.

Ginsburg, A. P. (1971). Psychological correlates of a model of the human visual system. In *Proceedings of the National Aerospace Electronics Conference (NAECON)* (pp. 283–290). Ohio: IEEE Trans. Aerospace Electronic Systems.

Ginsburg, A. P. (1978). Visual information processing based on spatial filters constrained by biological data. Aerospace Medical Research Laboratory, 1(2), Dayton, Ohio.

Ginsburg, A. P. (1980). Specifying relevant spatial information for image evaluation and display designs: An explanation of how we see certain objects. *Proceedings of SID, 21*, 219–227.

Ginsberg, A. P. & Evans, P. W. (1979). Predicting visual illusions from filtered imaged based on biological data. *Journal of the Optical Society of America A, 69*, 1443.

Harmon, L. D. & Julesz, B. (1973). Masking in visual recognition: Effects of two-dimensional filtered noise. *Science, 180*, 1194–1197.

Janez, L. (1984). Visual grouping without low spatial frequencies. *Vision Research, 24*, 271–274.

Kersten, D. (1984). Spatial summation in visual noise. *Vision Research, 24*, 1977–1990.

Legge, G. E., Pelli, D. G., Rubin, G. S. & Schleske, M. M. (1985). Psychophysics of reading—I. Normal vision. *Vision Research, 25(2)*, 239–252.

Legge, G. E., Kersten, D. & Burgess, A. E. (1987). Contrast discrimination in noise. *Journal of the Optical Society of America A, 4(2)*, 391–404.

van Meeteren, A. & Barlow, H. B. (1981). The statistical efficiency for detecting sinusoidal modulation of average dot density in random figures. *Vision Research, 21*, 765–777.

van Nes, F. L. & Bouman, M. A. (1967). Spatial modulation transfer in the human eye. *Journal of the Optical Society of America, 57*, 401–406.

Norman, J. & Ehrlich, S. (1987). Spatial frequency filtering and target identification. *Vision Research, 27(1)*, 97–96.

Parish, D. H. & Sperling, G. (1987a). Object spatial frequencies, retinal spatial frequencies, and the efficiency of letter discrimination. Mathematical Studies in Perception and Cognition, 87–8. New York University, Department of Psychology.

Parish, D. H. & Sperling, G. (1987b). Object spatial frequency, not retinal spatial frequency, determines identification efficiency. *Investigative Ophthalmology and Visual Science (ARVO Suppl.), 28(3)*, 359.

Pavel, M., Sperling, G., Riedl, T. & Vanderbeek, A. (1987). The limits of visual communication: The effect of signal-to-noise ratio on the intelligibility of American sign language. *Journal of the Optical Society of America A, 4*, 2355–2365.

Pelli, D. G. (1981). Effects of visual noise. Ph.D. dissertation, University of Cambridge, England.

Shannon, C. E. & Weaver, W. (1949). *The mathematical theory of communication*. Urbana: University of Illinois Press.

Sperling, G. (1989). Three stages and two systems of visual processing. *Spatial Vision, 4 (Prazdny Memorial Issue)*, 183–207.

Sperling, G. & Parish, D. H. (1985). Forest-in-the-Trees illusions. *Investigative Ophthalmology and Visual Science (ARVO Suppl.), 26*, 285.

Tanner, W. P. & Birdsall, T. G. (1958). Definitions of d' and n as psychophysical measures. *Journal of the Acoustical Society of America, 30*, 922–928.

van Tress, H. L. (1968). *Detection, estimation and modulation theory*. New York: Wiley.

Winer, B. J. (1971). *Statistical principles in experimental psychology*. New York: McGraw-Hill.

APPENDIX

Both sub-ideal and super-ideal discriminators must compute estimates of the likelihood that the stimulus $u_{k,b}$ was produced with template $t_{i,b}$ and noise n_b, where k is the letter used to generate the stimulus, i is an arbitrary letter, and b indexes spatial frequency band. Let x be an index on the pixels of the image: $1 \leq x \leq 8100$, for the 90×90 images of the experiments.

For the Monte Carlo simulations of the super-ideal discriminator, the unknown stimulus parameters, $\alpha_{i,b}$ and σ_N^2 are computed during stimulus construction, and their exact values are supplied to the discriminator *a priori*. The sub-ideal discriminator, however, must estimate these parameters from the data as follows.

Sub-Ideal Parameter Estimation

Recall that stimulus contrast is modulated for any pixel x in the image:

$$u_{k,b}[x] = \beta_{i,b}[t_{i,b}(x) + n_b(x)] + q_{i,b}(x). \quad \text{(A1)}$$

The scaling constant $\beta_{i,b}$ limits range of real values for each pixel, prior to quantization, to the open interval $(-0.5, 255.5)$; the addition of $q_{i,b}[x]$, called quantization noise, rounds off pixel values to integers.

For each bandpass filtered template $t_{i,b}$, we first compute the correlation $\rho_{k,i}$ of the template to the stimulus $u_{k,b}$:

$$\rho_{k,i} = \frac{\sum_x u_{k,b}(x)\, t_{i,b}(x)}{\left\{\sum_x [u_{k,b}(x)]^2\right\}^{1/2} \left\{\sum_x [t_{i,b}(x)]^2\right\}^{1/2}}. \quad \text{(A2)}$$

To compute the likelihood estimates for each template $t_{i,b}$, we must be able to reverse the effect of $\beta_{i,b}$. Thus we define $\alpha_{i,b} = 1/\beta_{i,b}$ and choose $\alpha_{i,b}$ so as to minimize the expression:

$$\sum_x [\alpha_{i,b} u_{k,b}(x)]^2 = \sum_x [\rho_{k,i} t_{i,b}(x)]^2. \quad \text{(A3)}$$

Solving for $\alpha_{i,b}$ gives us:

$$\alpha_{i,b} = \rho_{k,i} \left\{ \frac{\sum_x [t_{i,b}(x)]^2}{\sum_x [u_{k,b}(x)]^2} \right\}^{1/2}. \quad \text{(A4)}$$

Finally we set:

$$\sigma_N^2 = \frac{1}{X} \sum_{x=1}^{X} [\alpha_{i,b} u_{k,b}(x) - t_{i,b}(x)]^2 \quad \text{(A5)}$$

where $X = 8100$, the number of pixels in the image.

Likelihood Estimation

With estimates of σ_N^2 and $\alpha_{i,b}$ for the sub-ideal discriminator, and the *a priori* values for the super-ideal discriminator, we can formulate a maximum likelihood estimator. By rearranging terms of equation (A1) and dividing both sides by β yields:

$$\frac{u_{k,b}(x)}{\beta} - t_{i,b}(x) = n_b(x) + \frac{q_{i,b}(x)}{\beta}. \quad \text{(A6)}$$

Substituting $\alpha_{i,b}$ for $1/\beta$, and by transposing into the frequency domain, denoted by upper-case letters and indexed by ω, we have:

$$\alpha_{i,b} U_{k,b}(\omega) - T_{i,b}(\omega) = N_b(\omega) + \alpha_{i,b} Q_{i,b}(\omega). \quad \text{(A7)}$$

Note that the left side of equation (A7) is simply a difference image between the stimulus $U_{k,b}(\omega)$ and the template $T_{i,b}(\omega)$. This difference is exactly equal to the sum of the luminance and quantization noise only when the correct template is chosen ($i = k$). When the incorrect template is chosen ($i \neq k$) the right hand side of equation (A7) is equal to the sum of the noise sources plus some residue that is equal to $T_{k,b}(\omega) - T_{i,b}(\omega)$. Under the assumption that quantization noise can be modeled as independent additive noise in the frequency domain, the density Λ of the joint realization of the right-hand side of equation (A7) is given by:

$$\Lambda = \prod_\omega \frac{X}{\pi[\sigma_Q^2 \alpha_{i,b}^2 + \sigma_N^2 |F_b(\omega)|]^2} \times \exp\left[\frac{-X|\alpha_{i,b} U_{k,b}(\omega) - T_{i,b}(\omega)|^2}{\alpha_{i,b}^2 \sigma_Q^2 + \sigma_N^2 |F_b(\omega)|^2}\right] \quad \text{(A8)}$$

where $F_b(\omega)$ is simply the kernel of filter b, in the frequency domain. Dropping the multiplicative term in equation (A8), which does not depend on the template T, and taking logs, the ideal discriminator chooses the template that minimizes:

$$\sum_\omega \frac{X|\alpha_{i,b} U_{k,b}(\omega) - T_{i,b}(\omega)|^2}{\alpha_{i,b}^2 \sigma_Q^2 + \sigma_N^2 |F_b(\omega)|^2}. \quad \text{(A9)}$$

Finally, it is more convenient to compute the power of the quantization noise in the space domain (σ_q^2) than in the frequency domain (σ_Q^2): $\sigma_q^2 = \sigma_Q^2$. Spatial quantization noise, $q_{i,b}(x)$, is uniformly distributed on the interval $[-0.5, 0.5)$, so that σ_q^2 is computed as:

$$\int_{-0.5}^{0.5} x^2 \, dx \quad \text{(A10)}$$

and is equal to 1/12.

Visual Factors in Letter Identification

Denis G. Pelli, Catherine W. Burns, Manoj Raghavan, and Bart Farell
Institute for Sensory Research, Syracuse University, Syracuse, New York

We have been studying how people identify letters. Our results indicate that the process of letter identification is mediated by a general visual object recognition process.

Task

We briefly present a low contrast letter with independent Gaussian noise added to each pixel. Then the observer is shown a complete high-contrast alphabet and asked to indicate which letter was seen. An adaptive procedure adjusts the letter contrast on successive trials (each with independent noise) to estimate the "threshold" letter contrast at which the observer attains 62% correct.

Efficiency

For comparison, we also implement the ideal Bayesian classifier, using exactly the same task. "Efficiency" is the ratio of contrast energies at threshold (which is the squared ration of ideal to human threshold contrasts).

Alphabets

We have tested fluent readers of English, Devanagari (the script used for Hindi and Sanskit), Hebrew, and Armenian. The appearances of these alphabets are very different, yet their efficiencies are all about 10%.

Learning

We have measured the learning of new alphabets by observers of all ages. Learning proceeds at a similar rate, per trial, in all observers and alphabets, reaching expert performance (indistinguishable from a fluent reader) after a mere 3,000 trials. This includes a previously illiterate 3-year old learning the English alphabet, and adult readers learning foreign alphabets.

Novel Alphabets

We have created novel alphabets: two series of 26 random checkerboards. They are learned at similar rates as the traditional alphabets, but the asymptotic efficiencies are different. For a 4x4 checkerboard the efficiency is about 6%. For a 2x3 checkerboard the efficiency is about 24%. The similar fast learning rate for traditional and novel alphabets indicates that the process is not unique to reading, instead reflecting the operation of a general visual object recognition process.

Critical Band

Solomon and Pelli (1994) measured the effects of visual noise at various spatial frequencies on the threshold for letter identification. Their results reveal that letter identification is mediated by an octave-wide bandpass filter centered at 3 cycles per letter. The insensitivity at low spatial-frequencies confirms Parish and Sperling (1991).

References

1. Parish, D. H. and Sperling, G. (1991) Object spatial frequencies, retinal spatial frequencies, noise, and the efficiencies of letter discrimination. *Vision Research*, **31**, 1399-1416.
2. Solomon, J. A. and Pelli, D. G. (1994) The visual channel that mediates letter identification. Submitted to *Nature*.

An Improved Detection Model for DCT Coefficient Quantization

Heidi A. Peterson
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Albert J. Ahumada, Jr. and Andrew B. Watson
NASA Ames Research Center, Human Interface Research Branch
Moffett Field, California 94035-1000

ABSTRACT

A detection model is developed to predict visibility thresholds for discrete cosine transform coefficient quantization error, based on the luminance and chrominance of the error. The model is an extension of a previously proposed luminance-based model, and is based on new experimental data. In addition to the luminance-only predictions of the previous model, the new model predicts the detectability of quantization error in color space directions in which chrominance error plays a major role. This more complete model allows DCT coefficient quantization matrices to be designed for display conditions other than those of the experimental measurements: other display luminances, other veiling luminances, other spatial frequencies (different pixel sizes, viewing distances, and aspect ratios), and other color directions.

1. INTRODUCTION

1.1 Discrete cosine transform-based image compression

The discrete cosine transform (DCT) has become a standard method of image compression.[1,2,3] Typically the image is divided into 8×8-pixel blocks, which are each transformed into 64 transform coefficients. The DCT transform coefficients $I_{m,n}$, of an $N \times N$ block of image pixels $i_{j,k}$, are given by

$$I_{m,n} = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} i_{j,k}\, c_{j,m}\, c_{k,n}\,, \quad m,n = 0, \ldots, N-1, \tag{1a}$$

where

$$c_{j,m} = \alpha_m \cos\left(\frac{\pi m}{2N}[2j+1]\right) \quad , \quad \text{and} \quad \alpha_m = \begin{cases} \sqrt{1/N}\,, & m = 0 \\ \sqrt{2/N}\,, & m > 0 \end{cases} . \tag{1b}$$

The block of image pixels is reconstructed by the inverse transform:

$$i_{j,k} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} I_{m,n}\, c_{j,m}\, c_{k,n}\,, \quad j,k = 1, \ldots, N-1, \tag{2}$$

which for this normalization is the same as the forward transform. Quantization of the DCT coefficients achieves image compression, but also causes distortion in the decompressed image. Specifically, quantization of coefficient $I_{m,n}$ induces an error image which is simply the associated basis function, with amplitude equal to the coefficient quantization error (neglecting the DCT normalization).

1.2 The Quantization Matrix

The JPEG compression standard[1,2] requires that uniform quantizers be used for all the DCT coefficients. The quantizer step size used for each coefficient is determined by the user. A matrix is used to specify the quantization of the DCT coefficients, where the m,nth entry, $Q_{m,n}$, in the matrix gives the quantizer step size for coefficient $I_{m,n}$. Two example quantization matrices have been included in the JPEG standard. These

0-8194-1146-9/93/$6.00

matrices are given in Tables K.1 and K.2 of reference(2) and in Table 5 of reference(4). One of these matrices is commonly used for graylevel images, and for the luminance component image of color images; the other matrix is used for chrominance images. These matrices were designed for a particular compression/viewing scenario, and it is not clear how they should be changed when used under different viewing conditions, or especially for compression in a different color space. In this paper we propose a quantization matrix design technique that can be applied under a wide variety of conditions: different display luminances, veiling luminances, spatial frequencies, and color spaces.

2. DETECTION MODELS

2.1 Luminance-only Detection Model

Peterson, Peng, Morgan, and Pennebaker[4] developed quantization matrices for compressing images in the RGB color space (a different matrix is used for each of the R, G, and B component images). The matrices were derived from measured detection thresholds for small patches of replicated DCT basis functions, produced on a monitor using an individual R, G, or B gun on a black background. With minor adjustments, the measured thresholds were converted to quantization matrices which performed well in informal tests.

Ahumada and Peterson[5] proposed that the threshold measurements of Peterson *et al.*[4] could be predicted by a luminance-only detection model. The theoretical basis of their model is the assumption that the detectability of distortion in the decompressed RGB image can be predicted from the luminance contrast of the error image caused in a color component image by quantization of an individual DCT coefficient for a single block. That is, if the quantization error images associated with all the quantized DCT coefficients in all image blocks in all three color component images have amplitudes below their respective visibility thresholds, then no distortion will be visible in the decompressed image.

The Ahumada/Peterson luminance-only detection model approximates the log of the contrast sensitivity function (the dependence of the inverse threshold contrast on spatial frequency) by a parabola in log spatial frequency. The predicted log luminance threshold of the m,nth DCT basis function is

$$\log T_{L,m,n} = \log \frac{s\, b_L}{r_L + (1-r_L)\cos^2\theta_{m,n}} + k_L(\log f_{m,n} - \log f_L)^2, \qquad m,n = 0, \ldots, N-1. \tag{3}$$

The minimum luminance threshold, $s\ b_L$, occurs at spatial frequency f_L, and k_L determines the steepness of the parabola. The parameter $0.0 < s < 1.0$ is to account for visual system summation of quantization errors over a spatial neighborhood. Such spatial summation causes a decrease in threshold. The spatial frequency, $f_{m,n}$, associated with the m,nth basis function, is given by

$$f_{m,n} = \frac{1}{2N}\sqrt{(m/W_x)^2 + (n/W_y)^2}, \tag{4}$$

where W_x is the horizontal and W_y the vertical size of a pixel in degrees of visual angle. The model includes a factor ($r_L + (1-r_L)\cos^2\theta_{m,n}$) which accounts for the imperfect summation of the two Fourier components present in basis functions having two cosine components (m and $n \neq 0$), and also accounts for the reduced sensitivity due to the obliqueness of these Fourier components. The magnitude of the summation/obliqueness effect is determined by $0.0 < r_L < 1.0$, and the angular parameter $\theta_{m,n}$ is given by

$$\theta_{m,n} = \arcsin \frac{2 f_{m,0} f_{0,n}}{f_{m,n}^2}. \tag{5}$$

Based on a fourth power summation rule for the two Fourier components[5], r_L is set to 0.6. The oblique effect can be included by decreasing the value of r_L.

Ahumada and Peterson[5] fit this model to the Peterson *et al.*[4] threshold data, and then used the grating detection data of Van Nes and Bouman[6] to derive luminance dependencies for b_L, f_L, and k_L, thus enabling the to model be used for a range of viewing conditions affecting luminance, contrast, and spatial frequency of the quantization errors. Since the single gun measurements of Peterson *et al.*[4] mainly varied the intensity of the

spatial modulation (chrominance remained relatively constant), the ability of the luminance-only model to predict visibility thresholds for modulations in combined luminance and chrominance directions was not adequately tested. Also, the replicated DCT basis functions used by Peterson *et al.*[4] have Fourier transforms possibly more like those of grating studies than those of single basis functions[7,8]. To address these issues, Peterson[9] made new threshold measurements of single basis function, single monitor-gun test images superimposed on a white background (1931 CIE coordinates: $X_0 = 37.27$, $Y_0 = 41.19$, $Z_0 = 29.65$). This configuration gives test stimuli having more significant chrominance modulation. Figure 1 shows the new measured thresholds for basis functions where m or $n = 0$.

A parabola representing a version of the luminance-only model is also shown in Figure 1. This model does a fair job of predicting the measured thresholds independent of color direction, except for the DC (m and $n = 0$) thresholds, which are obviously different for the three color guns. We propose that the lower thresholds for the R and B gun DC basis functions are the result of chromatic detection mechanisms having greater sensitivity than the luminance mechanism. Thus, even for quantization in the RGB color space, a luminance-only model is not quite sufficient. Color mechanisms must be taken into account to determine appropriate quantization levels for the DC coefficients. More importantly, for images compressed using isoluminant color directions, a complete color space discrimination model for the DCT basis functions is clearly needed.

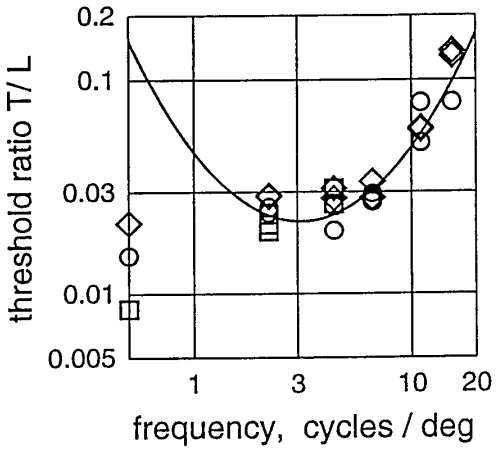


Figure 1: Visibility threshold contrast ratio measurements from Peterson[9] of single basis function, single monitor-gun test images superimposed on a white background, for basis functions where m or $n = 0$. Circles indicate R gun thresholds, diamonds indicate G gun thresholds, and squares indicate B gun thresholds. The points plotted at the far left of the graph are DC basis function (m and $n = 0$) thresholds. The parabola-shaped curve represents a version of the luminance-only model of Equation (3).

2.2 The Luminance/Chrominance Detection Model

To account for the DC sensitivities in the data of Figure 1, we add two chromatic channels to the luminance-only model. A large number of different color spaces have been proposed as appropriate bases for chromatic discriminations. We have selected for our chrominance channels those favored by Boynton[10]: a red-green opponent channel and a blue channel. The relation between these chromatic channels and the CIE 1931 XYZ color space is straightforward. The blue channel is just Z, and the red-green opponent channel O is given by $O = 0.47X - 0.37Y - 0.10Z$. This opponent channel is Boynton's[10] (Red-cone) − 2(Green-cone) channel, with the Red and Green XYZ cone responses taken from MacLeod and Boynton[10]. (We ignore the small correction developed by Vos[14] for going from the 1931 standard CIE values to the scientifically favored 1951 Judd CIE values used by MacLeod and Boynton.) Expressed in matrix form, the transformation from XYZ to

our YOZ opponent color space is

$$[\ YOZ\ ] = [\ XYZ\ ]_{XYZ}M_{YOZ} = [\ XYZ\ ]\begin{bmatrix} 0 & 0.47 & 0 \\ 1 & -0.37 & 0 \\ 0 & -0.10 & 1 \end{bmatrix}. \tag{6}$$

We model the frequency response of the Y channel with the luminance-only model described above. To reflect this, we subsequently refer to threshold $T_{L,m,n}$ as $T_{Y,m,n}$. The parameters in the luminance channel model will subsequently be referred to with a similar change of subscript (L → Y). To complete our luminance/chrominance model, we must also specify the shape of the frequency responses of the O and Z channels. Measurements of the spatial frequency responses of isoluminant chromatic modulations have typically found the chromatic sensitivity functions (the dependence of the inverse threshold contrasts on spatial frequency) to be low-pass in the frequency range of our basis functions and to be less sensitive at high spatial frequencies than the luminance channel.[11,12] We therefore model each of the O and Z log chromatic thresholds as a parabola, modified by setting it equal to its minimum value for all spatial frequencies to the left of the minimum. Since the data of Peterson[9] are too sparse to estimate two separate chromatic channels in close proximity, we make the simplifying assumption, supported by the results of Mullen[11], that both O and Z have the same shape spatial frequency response. The O and Z log chromatic thresholds for the m,nth DCT basis function can then be written:

$$\log T_{O,m,n} = \begin{cases} \log \dfrac{s\, b_O}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}}, & \text{if } f_{m,n} \le f_{OZ} \\ \log \dfrac{s\, b_O}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}} + k_{OZ}(\log f_{m,n} - \log f_{OZ})^2, & \text{if } f_{m,n} > f_{OZ} \end{cases}. \tag{7a}$$

and

$$\log T_{Z,m,n} = \begin{cases} \log \dfrac{s\, b_Z}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}}, & \text{if } f_{m,n} \le f_{OZ} \\ \log \dfrac{s\, b_Z}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}} + k_{OZ}(\log f_{m,n} - \log f_{OZ})^2, & \text{if } f_{m,n} > f_{OZ} \end{cases}. \tag{7b}$$

Note that Equations (7a) and (7b) are identical, except for the parameters b_O and b_Z; $T_{O,m,n}$ and $T_{Z,m,n}$ share the parameters s, k_{OZ}, f_{OZ}, and r_{OZ}. To obtain the overall model threshold $T_{m,n}$ from the three channel thresholds, we use the "minimum of" combination rule:

$$T_{m,n} = \min\{\ T_{Y,m,n},\ T_{O,m,n},\ T_{Z,m,n}\ \}. \tag{8}$$

In order to estimate the parameters in the model described above, we fit the model to the data of Peterson[9] shown in Figure 1. Recall that the Peterson[9] thresholds were measured for single basis functions. To reflect the absence of a spatial summation effect in this data, we fixed $s = 1.0$ during the fitting process. This fit resulted in the parameter values shown in Table 1 for k_Y, f_Y, k_{OZ}, and f_{OZ}. We chose $r_{OZ}=0.6$, the same as r_Y.

Boynton[10] claims that at moderately high intensities, the Z channel's minimum threshold ($s\, b_Z$ in our model) is approximately proportional to the background activity of the Z channel, and the minimum thresholds for the Y and O channels ($s\, b_Y$ and $s\, b_O$ in our model) are approximately proportional to the background Y. Based on the fit of our model to the Figure 1 data, we set the constants of proportionality to be: $b_Y = 0.0219\, Y_0$, $b_O = 0.0080\, Y_0$, and $b_Z = 0.0647\, Z_0$, where Y_0 and Z_0 are the CIE values of average white. To determine a value for s, we compared the thresholds measured in Peterson[9] to those measured by Van Nes and Bouman[6] for large test pattern sinusoidal gratings. The Peterson[9] thresholds are consistently higher than the Van Nes and Bouman[6] thresholds, a result attributable to spatial summation. Multiplication of the Peterson[9] data by 0.25 brings them into approximate agreement with the Van Nes and Bouman[6] data. We therefore chose $s = 0.25$. These results are summarized in Table 1.

Table 1. Parameter values estimated for the model of Equation (8).

model channel	parameter values				
	s	r	f	k	b
Y	0.25	0.6	3.1	1.34	$0.0219\,Y_0$
O	0.25	0.6	1.0	3.00	$0.0080\,Y_0$
Z	0.25	0.6	1.0	3.00	$0.0647\,Z_0$

As part of the model fitting, we also tried the Euclidean distance combination rule:

$$T_{m,n}^{-2} = T_{Y,m,n}^{-2} + T_{O,m,n}^{-2} + T_{Z,m,n}^{-2}. \tag{9}$$

However, when the data of Figure 1 were fit using this rule, in order to prevent contributions from the chromatic channels at low spatial frequencies, f_{OZ} was forced to be unrealistically low, and/or k_{OZ} was forced to be unrealistically high. This led to our selection of the "minimum of" rule for $T_{m,n}$.

3. QUANTIZATION MATRIX DESIGN

Quantization errors in an arbitrary color space are interpreted in the following way. Suppose we wish to compress a color image whose pixels are computed as a linear combination of XYZ values,

$$[\mathrm{DEF}] = [\mathrm{XYZ}]\ {}_{\mathrm{XYZ}}M_{\mathrm{DEF}}. \tag{10}$$

That is, the DCT is to be performed on an image in color space DEF, and ${}_{\mathrm{XYZ}}M_{\mathrm{DEF}}$ is the transformation from XYZ to DEF. The image in DEF space can be thought of as being transformed to XYZ space, and then converted by the visual system to YOZ space for discrimination. We need to determine limits on the sizes of errors in each of the D, E, and F color space dimensions, in order for the resulting errors in the Y, O, and Z channels to all be below the thresholds established by our model. These DEF thresholds determine the quantization matrices. For example, a unit error in the amplitude of a DCT coefficient in dimension D induces errors whose amplitudes in the Y, O, and Z channels are given by the first row of ${}_{\mathrm{DEF}}M_{\mathrm{YOZ}}$:

$${}_{\mathrm{DEF}}M_{\mathrm{YOZ}} = {}_{\mathrm{DEF}}M_{\mathrm{XYZ}} \times {}_{\mathrm{XYZ}}M_{\mathrm{YOZ}} = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}, \tag{11}$$

where ${}_{\mathrm{DEF}}M_{\mathrm{XYZ}}$ is the inverse of ${}_{\mathrm{XYZ}}M_{\mathrm{DEF}}$.

We now describe in detail the procedure to calculate $Q_{D,m,n}$, $Q_{E,m,n}$, and $Q_{F,m,n}$, the quantization matrix entries for DCT coefficient $I_{m,n}$ in the D, E, and F component images. First, using Equations (3) and (7), the display parameters W_x and W_y, and the model parameters given in Table 1, the model channel thresholds, $T_{Y,m,n}$, $T_{O,m,n}$, and $T_{Z,m,n}$, for the m,nth DCT basis function are calculated. Now let ${}_{Y}T_{D,m,n}$, ${}_{O}T_{D,m,n}$, and ${}_{Z}T_{D,m,n}$ indicate the thresholds imposed on the quantization error in the D component by the model's thresholds for the Y, O, and Z channels, respectively. Each of the Y, O, and Z model channel thresholds are converted to a D threshold as follows:

$${}_{Y}T_{D,m,n} = \frac{T_{Y,m,n}}{|M_{1,1}|}, \quad {}_{O}T_{D,m,n} = \frac{T_{O,m,n}}{|M_{1,2}|}, \text{ and } \quad {}_{Z}T_{D,m,n} = \frac{T_{Z,m,n}}{|M_{1,3}|}. \tag{12a}$$

Similarly for E and F:

$${}_{Y}T_{E,m,n} = \frac{T_{Y,m,n}}{|M_{2,1}|}, \quad {}_{O}T_{E,m,n} = \frac{T_{O,m,n}}{|M_{2,2}|}, \quad {}_{Z}T_{E,m,n} = \frac{T_{Z,m,n}}{|M_{2,3}|}, \tag{12b}$$

$${}_{Y}T_{F,m,n} = \frac{T_{Y,m,n}}{|M_{3,1}|}, \quad {}_{O}T_{F,m,n} = \frac{T_{O,m,n}}{|M_{3,2}|}, \quad {}_{Z}T_{F,m,n} = \frac{T_{Z,m,n}}{|M_{3,3}|}. \tag{12c}$$

Then the combination rule is used to determine the D, E, and F thresholds. We use the "minimum of" rule:

$$T_{D,m,n} = \min\{\ {}_{Y}T_{D,m,n},\ {}_{O}T_{D,m,n},\ {}_{Z}T_{D,m,n}\ \}, \tag{13a}$$

$$T_{E,m,n} = \min\{\ {}_{Y}T_{E,m,n},\ {}_{O}T_{E,m,n},\ {}_{Z}T_{E,m,n}\ \}, \tag{13b}$$

$$T_{F,m,n} = \min\{\ {}_{Y}T_{F,m,n},\ {}_{O}T_{F,m,n},\ {}_{Z}T_{F,m,n}\ \}. \tag{13c}$$

Finally, the D, E, and F quantization matrix entries are obtained by dividing the thresholds above by the DCT normalization constants (α_m in Equation (1b)):

$$Q_{D,m,n} = 2\frac{T_{D,m,n}}{\alpha_m \alpha_n}, \qquad Q_{E,m,n} = 2\frac{T_{E,m,n}}{\alpha_m \alpha_n}, \qquad Q_{F,m,n} = 2\frac{T_{F,m,n}}{\alpha_m \alpha_n}. \tag{14}$$

The factor 2 results from the maximum quantization error being half the quantizer step size.

3.1 Quantization in RGB space

For quantization in monitor-RGB space, we require the matrix to transform from RGB to XYZ space, ${}_{RGB}M_{XYZ}$. Assuming that R, G, and B take on values between 0 and 1, ${}_{RGB}M_{XYZ}$ is the monitor calibration matrix giving the XYZ values for unit changes in each of the RGB signals. For our monitor,

$${}_{RGB}M_{XYZ} = \begin{bmatrix} 26.1 & 13.3 & 2.3 \\ 25.2 & 48.9 & 10.2 \\ 9.3 & 4.7 & 35.7 \end{bmatrix}. \tag{15}$$

This matrix is post-multiplied by ${}_{XYZ}M_{YOZ}$ to obtain ${}_{RGB}M_{YOZ}$:

$$[\ YOZ\ ] = [\ RGB\ ]\ {}_{RGB}M_{YOZ} = [\ RGB\ ] \begin{bmatrix} 13.3 & 7.1 & 2.3 \\ 48.9 & -7.3 & 10.2 \\ 4.7 & -0.9 & 35.7 \end{bmatrix}. \tag{16}$$

The matrix ${}_{RGB}M_{YOZ}$ gives the amplitude of the YOZ errors resulting from unit errors in RGB. These values indicate the sensitivity of the discrimination model YOZ channels to RGB errors. For example, a unit error in the R component leads to an error of 7.1 in the O channel of the model.

We can calculate the R, G, and B coordinate increments which induce a minimum threshold step in each of the Y, O, and Z channels. These are the the entries of ${}_{RGB}M_{YOZ}$ divided into the appropriate minimum threshold: $s\,b_Y$, $s\,b_O$, or $s\,b_Z$, calculated using the expressions in Table 1 and the Y_0 and Z_0 values of our average white. For example, letting $({}_{RGB}M_{YOZ})_{1,1}$, signify the upper left corner entry in ${}_{RGB}M_{YOZ}$, the increment in R which results in a minimum threshold change in Y is $(s\,b_Y) / ({}_{RGB}M_{YOZ})_{1,1}$. RGB minimum threshold increments calculated in this way are given in Table 2 for YOZ. Note that the minimum threshold for G is determined by the Y channel (0.0046 versus 0.0113 and 0.0469). That is, the Y channel imposes the strictest limit on G in order for a G change to not induce "too large" a change in YOZ-space. Similarly, the minimum threshold for R comes from the O channel (0.0116), and for B comes from the Z channel (0.0134). Following the procedure described above, using ${}_{RGB}M_{YOZ}$, the model parameters in Table 1, and the Y_0 and Z_0 values for our monitor, we obtain the quantization matrices shown in Table 3 for our RGB color space.

Table 2. Minimum thresholds imposed on R, G, and B quantization errors by the Y, O, and Z model minimum thresholds.

	Y	O	Z
R	0.0170	0.0116	0.2091
G	0.0046	-0.0113	0.0469
B	0.0483	-0.0881	0.0134

Table 3. RGB quantization matrices. The values in these matrices are obtained following the procedure described in Section 3. The $Q_{0,0}$ value is located in the upper left corner of each quantization matrix. As specified in the JPEG standard, the values have been rounded to the nearest integer. JPEG also requires that values in the quantization matrix be ≤ 255.

R quantization matrix	47	52	53	69	94	127	170	224
	52	57	53	60	75	98	128	167
	53	53	77	89	103	124	154	192
	69	60	89	119	142	166	197	236
	94	75	103	142	181	217	254	297
	127	98	124	166	217	269	320	373
	170	128	154	197	254	320	388	457
	224	167	192	236	297	373	457	544
G quantization matrix	19	14	14	19	26	35	46	61
	14	16	14	16	21	27	35	45
	14	14	21	24	28	34	42	52
	19	16	24	32	39	45	54	64
	26	21	28	39	49	59	69	81
	35	27	34	45	59	73	87	102
	46	35	42	54	69	87	106	124
	61	45	52	64	81	102	124	148
B quantization matrix	55	94	151	197	268	363	486	641
	94	164	151	171	216	281	367	477
	151	151	221	254	294	355	440	550
	197	171	254	340	406	475	562	675
	268	216	294	406	519	621	727	851
	363	281	355	475	621	770	915	1066
	486	367	440	562	727	915	1109	1306
	641	477	550	675	851	1066	1306	1556

Figure 2 plots all the measured R, G, and B gun, single basis function thresholds from Peterson[9] (including those for the dual frequency (m and $n \neq 0$) basis functions), after correction by the summation/obliqueness factors of Equations (3) and (7). Figure 2 also shows the curves for the model threshold predictions $T_{Y,m,n}$, $T_{O,m,n}$, and $T_{Z,m,n}$, using the parameters in Table 1, except with $s = 1.0$. This value for s was used to reflect the absence of a spatial summation effect in the single basis function data. In addition, the $T_{O,m,n}$ and $T_{Z,m,n}$ threshold prediction curves have been converted to luminance units, since all the threshold data plotted are in luminance units. This is accomplished by multiplying the $T_{O,m,n}$ threshold predictions by 13.3 / 7.1, and the $T_{Z,m,n}$ threshold predictions by 4.7 / 35.7. These factors are obtained from the ${}_{RGB}M_{YOZ}$ matrix. Figure 2 shows that for the B component, the DC and lowest spatial frequency thresholds are determined by the Z channel, and for the R component, the DC threshold is determined by the O channel. All the G thresholds are assumed to be determined by the Y channel. Note that the DC threshold for the Y channel (which we assume to be the DC threshold measured for G) is not predicted on a theoretical basis. The dot-dashed line in Figure 2 demonstrates that the measured DC threshold for G, and hence our DC threshold for Y, was found to be approximately equal to the minimum threshold of the Y channel.

4. CONCLUSIONS

We have presented a model for predicting visibility thresholds for DCT coefficient quantization error, from which quantization matrices for use in DCT-based compression can be designed. We estimated values for the parameters of our model based on experimentally measured visibility thresholds. The frequency parameters we estimated, f_Y and f_{OZ}, agree fairly well with results others have reported for similar parameters. The values we have estimated for k_Y and k_{OZ} are similar to those estimated by others, however we have found these parameters to vary for the different experimentally measured thresholds. The value we have proposed for the obliqueness/summation parameters, r_Y and r_{OZ}, only reflects summation and does not reflect an effect due to obliqueness. More data may be needed to more determine values for k_Y, k_{OZ}, r_Y, and r_{OZ} more reliably; though those we propose here are reasonable and result in quantization matrices which perform well in preliminary tests. The value for s we have proposed is based on a limited amount of data. Further experiments are needed to determine the spatial extent over which summation occurs among DCT quantization errors, in order to estimate s more accurately.

The quantization matrices computed by the techniques described above take no account of image content. A promising extension of this model may be to optimize the quantization matrices for individual images or a class of images. That is, use an image-dependent approach to quantization matrix design. Watson[15] has shown how this may be done for grayscale images, by taking into account local light adaptation, local contrast masking, and error pooling. Watson's technique can be extended to the case of color images by adopting rules governing masking and adaptation within the O and Z channels.

5. ACKNOWLEDGMENTS

We appreciate the help of Jeffrey B. Mulligan. This work was supported in part by the IBM Independent Research and Development Program and by NASA RTOP Nos. 506-59-65 and 505-64-53.

REFERENCES

1. G. Wallace, "The JPEG still picture compression standard", *Communications of the ACM*, vol. 34, no. 4, pp. 30-44, 1991.
2. W. B. Pennebaker, J. L. Mitchell, *JPEG Still Image Data Compression Standard*, van Nostrand Reinhold, New York, 1993.
3. D. LeGall, "MPEG: A video compression standard for multimedia applications", *Communications of the ACM*, vol. 34, no. 4, pp. 46-58, 1991.
4. H. A. Peterson, H. Peng, J. H. Morgan, W. B. Pennebaker, "Quantization of color image components in the DCT domain", in B. E. Rogowitz, M. H. Brill, J. P. Allebach, eds., *Human Vision, Visual Processing, and Digital Display II*, Proc. SPIE, vol. 1453, pp. 210-222, 1991.
5. A. J. Ahumada, Jr., H. A. Peterson, "Luminance-model-based DCT quantization for color image compression," in B. E. Rogowitz, ed., *Human Vision, Visual Processing, and Digital Display III*, Proc. SPIE, vol. 1666, pp. 365-374, 1992.
6. F. L. van Nes, M. A. Bouman, "Spatial modulation transfer in the human eye", *Journal of the Optical Society of America*, vol. 57, pp. 401-406, 1967.
7. R. J. Clarke, "Spectral response of the discrete cosine and Walsh-Hadamard transforms," *IEEE Proceedings*, vol. 130, pp. 309-313, 1983.
8. S. A. Klein, A. D. Silverstein, T. Carney, "Relevance of human vision to JPEG-DCT compression," in B. E. Rogowitz, ed., *Human Vision, Visual Processing, and Digital Display III*, Proc. SPIE, vol. 1666, pp. 200-215, 1992.
9. H. A. Peterson, "DCT basis function visibility thresholds in RGB space," in J. Morreale, ed., *1992 SID International Symposium Digest of Technical Papers*, Society for Information Display, Playa del Rey, CA, pp. 677-680, 1992.

10. R. M. Boynton, *Human Color Vision*, Holt, Rinehart, and Winston, New York, 1979
11. K. T. Mullen, "The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings," *Journal of Physiology*, vol. 359, pp. 381-400, 1985.
12. D. H. Kelly, "Spatio-temporal frequency characteristics of color-vision mechanisms," *Journal of the Optical Society of America*, vol. 64, pp. 55-72, 1974.
13. D. I. A. MacLeod, R. M. Boynton, "Chromaticity diagram showing cone excitation by stimuli of equal luminance," *Journal of the Optical Society of America*, vol. 69, pp. 1183-1186, 1979.
14. J. J. Vos, "Colorimetric and photometric properties of a 2° fundamental observer," *Color Research and Application*, vol. 3, pp. 125-128, 1978.
15. A. B. Watson, "DCT quantization matrices visually optimized for individual images," in B. E. Rogowitz, J. P. Allebach, eds., *Human Vision, Visual Processing, and Digital Display IV*, (SPIE, Bellingham, WA, 1993).

Journal of Electronic Imaging 1(3), 328-338 (July 1992).

Human visual sensitivity-weighted progressive image transmission using the lapped orthogonal transform

Ricardo L. de Queiroz
K. R. Rao
University of Texas at Arlington
Electrical Engineering Department
P.O. Box 19016
Arlington, Texas 76019

Abstract. *Progressive transmission of images based on the lapped orthogonal transform (LOT), adaptive classification, and human visual sensitivity (HVS) weighting is proposed. HVS weighting for LOT basis functions is developed. This technique is quite general and can be applied to any orthogonal transform. The method is compared with discrete cosine transform (DCT)-based progressive image transmission (PIT). It is shown that LOT-based PIT yields subjectively improved images compared to those based on DCT. This is consistent with the reduction in block structure characteristic of LOT image coding.*

1 Introduction

While progressive image transmission[1] (PIT) can be classified into two major categories, i.e., (1) spatial or pel domain and (2) transform or spectral domain, the latter has gained wide acceptance.[2-10] This is not only due to various adaptive features such as classification,[11-16] spectral selection,[4,7,8] and human visual system (HVS) weighting,[2,7,8,17-21] etc., which can be easily incorporated into the transform coding scheme, but is also due to the VLSI development of coding operations such as transform, quantization, and variable length coding. In addition, PIT based on the discrete cosine transform (DCT) has been extensively investigated. For example, the JPEG (Joint Photographic Experts Group) algorithm[7,8] for the baseline system is DCT based and various hardware/software systems have already been developed for this algorithm. Also, the nonhierarchical extended system of JPEG (both spectral selection and successive approximation) is DCT based. At low bit rates, however, DCT introduces block structure in the reconstructed images.[2] One technique used to reduce or eliminate this artifact is to replace DCT by the lapped orthogonal transform (LOT),[22-28] whose basis vectors overlap across traditional block boundaries. Also because LOT has good filtering properties, it has been applied to compatible coding,[29,30] i.e., coding of the original image/sequence at different spatial resolutions. It has also been combined with vector quantization (VQ) to achieve additional compression.[31] It is intuitively felt that LOT-based PIT should yield subjectively more pleasing pictures compared to the DCT—even during the initial stages. This is the objective of this paper: to develop a LOT-PIT incorporating various adaptive features and to compare it with the DCT-dependent PIT.

In Sec. 2, we will address the Chen-Smith coder, giving a brief summary of the algorithm steps and explaining the incorporation of PIT techniques in this algorithm. Section 3 is reserved for a discussion about the HVS model in the transform domain. Simulations and coder details are presented in Sec. 4, with conclusions given in Sec. 5.

2 PIT with the Chen-Smith Coder

The Chen-Smith coder[12] is based on the zonal sampling strategy. First, the image undergoes an orthogonal transform. The transform coefficients are stored in a buffer and some statistics are computed prior to the decision-making process of (1) which coefficients are transmitted, (2) how these coefficients are quantized, and (3) the order of transmission. We will assume the image has $N \times N$ picture elements (pixels or pels).

The encoding steps can be briefly described as follows:

Paper 92-018 received April 7, 1992; revised manuscript received July 13, 1992; accepted for publication July 16, 1992.
1017-9909/92/$2.00.

- Transform the image using blocks of $M \times M$ pels. Let $N_B = (N/M)^2$ be the total number of blocks in the image. To simplify the presentation, we will use a lexicographic ordering that can obey row or column arrangement. The blocks are then labeled from 1 to N_B. Each one contains M^2 coefficients given as $x_i(u,v)$ for $i = 1, \ldots, N_B$ and $(u,v) \in \{(0,0), \Psi\}$, where Ψ is defined as the set of $M^2 - 1$ block-index pairs, excluding the pair (0,0), as $\Psi \equiv \{(0,1), (0,2), \ldots, (0, M-1), (1,0), (1,1), \ldots, (M-1, M-1)\}$.
- Quantize and code separately the coefficients $x_i(0,0)$ (the dc coefficients) using uniform quantizers.
- Compute the ac energy of each block E_i as

$$E_i = \sum_{(m,n) \in \Psi} x_i^2(m,n) \ . \tag{1}$$

Sort the energies, and classify the blocks (in sorted order) into N_C equally populated classes.[12] Thus, there would be N_B/N_C blocks in each class. Construct the *class map* $C(i)$ with the classification of each block, where $C(i)$ indicates the class to which the i'th block belongs and is ordered in the original nonsorted sequence. If the i'th block belongs to the class k ($k = 1, \ldots, N_C$), then $C(i) = k$.

- For all blocks belonging to the same class, compute the variances of the transform coefficients and then their standard deviations. Construct N_C *standard deviation maps* with the standard deviations of the coefficients, which are obtained from

$$\sigma_k^2(m,n) = \sum_{i=1}^{N_B} \delta[C(i) - k] x_i^2(m,n) \quad (m,n) \in \Psi \ , \tag{2}$$

where δ is the Kronecker delta function.

- Merge all N_C standard deviation maps and decide the bit allocation. Based on the rate-distortion theory, we shall iteratively find a distortion value D and a set of integers $B_k(m,n)$ [for $(m,n) \in \Psi$ and $1 \leq k \leq N_C$], so that

$$B_k(m,n) = \tfrac{1}{2} \log_2[\sigma_k^2(m,n)] - \log_2(D) \tag{3}$$

is satisfied given the constraints

$$\sum_{k=1}^{N_C - 1} \sum_{(m,n) \in \Psi} B_k(m,n) = (RN^2 - B_{ov}) \frac{N_C}{N_B} \ , \tag{4}$$

$$0 \leq B_k(m,n) \leq B_{max} \ , \tag{5}$$

where B_{max} is the maximum number of bits allowed, B_{ov} is the number of bits required for the transmission of the overhead information, and R is the bit rate in bits/pel for the whole image. Create N_C bit-allocation maps with a one-to-one correspondence with the elements of the standard deviation maps.

- Reestimate the standard deviations using the bit-allocation maps:

$$\hat{\sigma}_k(m,n) = c 2^{B_k(m,n) - 1} \quad 1 \leq k \leq N_C \quad (m,n) \in \Psi \ , \tag{6}$$

where c is a normalization factor. Reference 12 suggested that c be chosen as the maximum $\sigma_k(m,n)$ for which $B_k(m,n) = 1$ to avoid excessive clipping.

- Send class map c and the bit-allocation maps as side information.
- Quantize, encode, and send all the coefficients, using the reestimated variances. A coefficient $x_i(m,n)$ (block i), which belongs to class $k[C(i) = k]$, is scaled [divided by $\hat{\sigma}_k(m,n)$], applied to a quantizer with $2^{B_k(m,n)}$ levels, and encoded with $B_k(m,n)$ bits. If $B_k(m,n) = 0$, the particular coefficient is not transmitted.

The receiver may first decode the side information and the dc coefficients. Given the class map, the bit-allocation maps, and the normalization factor c, the decoder can reconstruct the standard deviations used to scale the quantizers as in Eq. (6). With the maps reconstructed, and with the knowledge of the transmission order, the decoder can exactly determine the position of the incoming coefficient, the class of its block, how many bits were assigned to it, and the variance used for quantization. Therefore, the receiver can decode the coefficients, apply an inverse transform, and obtain the image.

The overhead is made by the class map, the bit-allocation maps, and by c. Quantizing c with 16 bits, the total amount of overhead is given by:

$$B_{ov} = N_B \log_2(N_C) + N_C(M^2 - 1) \times \log_2(B_{max} + 1) + 16 \ . \tag{7}$$

If $M = 8$, $N = 256$, $N_C = 8$, $B_{max} = 7$, then $B_{ov} = 4552$, which is equivalent to an approximate rate of 0.07 bit/pel, requiring about 2 s of transmission on a 2400 bits/s communication rate.

To use PIT, we transmit data in the following order: (1) dc coefficients in any predefined order, (2) class map c and bit-allocation maps, (3) ac coefficients. The transmission of the ac coefficients[2] is made by spanning the blocks and sending first the elements $x_i(m,n)$, which would yield a higher contribution to the reconstructed image. To minimize the reconstruction error, we send the coefficients with higher variances. Alternatively, we can incorporate some information about the spatial response of the visual system, by using weighted standard deviations. If one assumes that the estimated standard deviation is a good measure of the real standard deviation of a particular coefficient (at least, is the best information we have at hand), the priority can be decided based on the weighting of the standard deviation maps by a matrix $\mathbf{H}(m,n)$ containing spatial information about the HVS. Let

$$\eta_k(m,n) = \hat{\sigma}_k(m,n) \mathbf{H}(m,n) \ ; \quad 1 \leq k \leq N_C; \quad (m,n) \in \Psi \ . \tag{8}$$

The order for transmission of the coefficients is then defined by sending first the coefficients $[x_i(m,n); C(i) = k]$, which correspond to: (1) greater value of $\eta_k(m,n)$; (2) if two or more $\eta_k(m,n)$ have the same value, take the one with smaller value of $m+n$; or (3) if there is still any ambiguity, take the smaller value of k.

The first item is the only one that follows any theoretical explanation; the last two are included merely for eliminating ambiguities, such as two equal values, and can be changed without affecting the performance. Note that using Eq. (6),

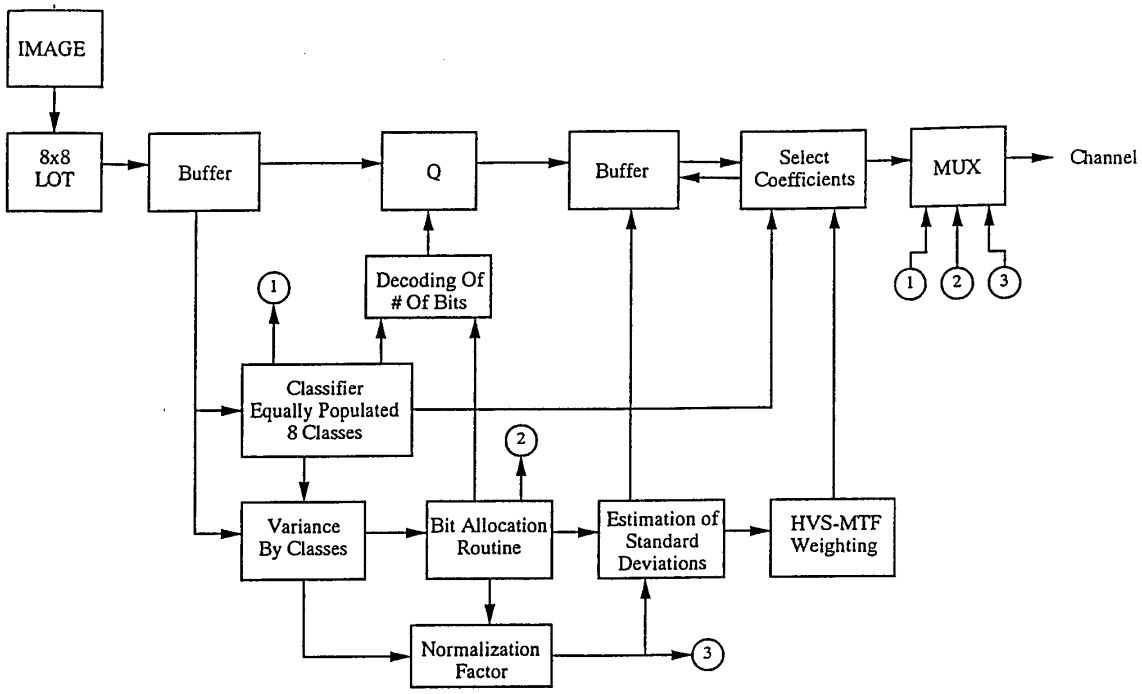


Fig. 1 Coder diagram for PIT using LOT.

we can take the log and sum $\log(2/c)$ on both sides of Eq. (8) so that $\eta_k(m,n)$ can be redefined as

$$\eta_k(m,n) = B_k(m,n) + \log_2[\mathbf{H}(m,n)] \ . \tag{9}$$

Although having a different value, this representation still maintains the transmission order, addressing directly the bit-allocation maps. As long as both encoder and decoder have the same maps and use the same weighting matrix, there will be no overhead for indicating the transmission order.

The coder has some limitations. First, the maximum number of different variances used for scaling the quantizers is $B_{\max}$. For high rates (>4 bits/pel), the performance decreases, since there will no longer be coefficients with only a few bits allocated. Second, it is not possible to apply HVS weighting to quantization without causing excessive mismatch or amplification of distortion because of the reestimation procedure in Eq. (6). It can be overcome by the transmission of standard deviations in place of the bit-allocation maps. We are interested in "small" pictures, such as 256×256 pel images. For these types of images, using 8 or 16 classes, the overhead for fully transmitting the variance maps would be prohibitive. The performance of this coder can be improved in several ways. For example, by choosing the proper parameters (block size, number of classes, and bit rate), the coder can achieve very good performance. The great advantage of the Chen-Smith approach is that it is quite insensitive to the transform used. One can interchangeably use DCT, LOT, extended lapped transforms,[27] or any transform resulting in blocks of $M \times M$ coefficients without any alteration in the algorithm (except for the weighting matrix and, possibly, coding details). This is the main reason for choosing the Chen-Smith coder.

The coder and decoder block diagrams employing the LOT are presented in Figs. 1 and 2, respectively.

3 The HVS Weighting Matrix

A complete study of the psychophysical properties of the visual system is well beyond the scope of this paper. Our intention is restricted to the determination of a spatial response weighting matrix for use with the LOT coefficients. We now present a procedure that allows us to find a HVS weighting function for any transform.

Reference 2 discussed the application of a linear function describing the HVS to spatial variations. Although the HVS model response is not linear, this principle was used with good results and further discussion on the subject is left to Ref. 2. Given a linear transfer function representing the unidimensional spatial HVS as $H(f)$ (where f is given in cycles per degree of the visual angle subtended), we will assume this model to be reliable and it will serve as the basis for the rest of this section. However, we will present our results as a function of the model in order to allow one to change $H(f)$ if desired. Further, the usual assumptions follow:

- The screen has a 1:1 ratio and has uniform brightness when displaying a uniform image.
- The viewer is situated at a distance v from the screen, right in front of its geometric center.

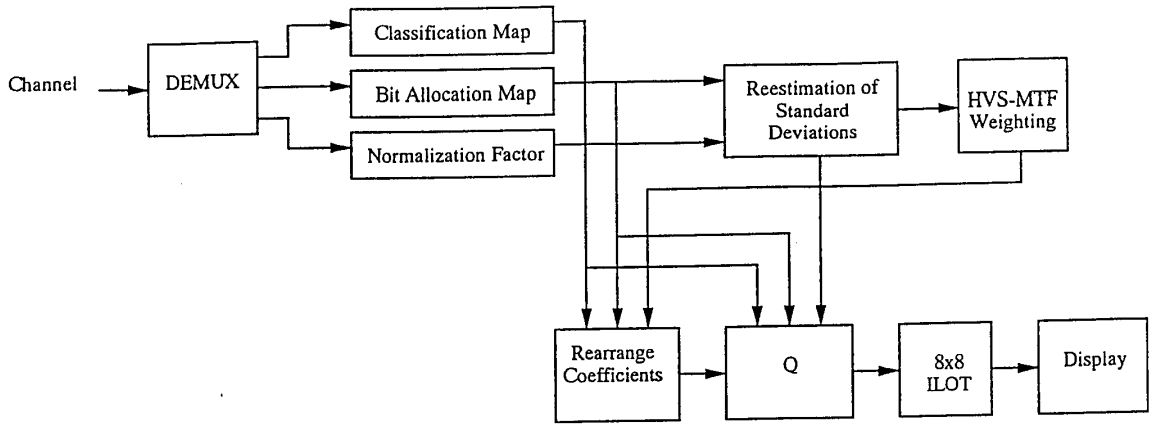


Fig. 2 Decoder diagram for PIT using LOT.

- The screen has width w and each row (column) has N pels.
- The viewer can observe approximately the same density of pels-per-degree (spatial) in any region of the screen.

Let α be the ratio of viewer distance (v) by screen width (w), i.e., $\alpha \equiv v/w$. This factor is the relative distance of the observer. The maximum visible frequency in cycles per degree is obtained when the discrete signal displayed has its maximum frequency component, which is half of the sampling frequency. In other words, in N samples it is possible to observe $N/2$ cycles. The maximum visible frequency can be found as:

$$f_{\max} = \frac{N/2}{2\theta} = \frac{N}{4 \arctan\left(\frac{1}{2\alpha}\right)} \text{ cycles/degree} , \tag{10}$$

where θ in degrees is the viewing angle, from the center to the extreme of the screen, and $\tan(\theta) = w/2v = 1/2\alpha$. We, therefore, can represent a discrete sensitivity function as

$$H_D(e^{j\omega}) = H_D(e^{j2\pi f}) = H(f/f_{\max}) \; ; \quad |f| < f_{\max} . \tag{11}$$

An orthogonal block transform is a special case of a lapped transform in which there are as many basis functions as elements in each basis function.[26] Furthermore, lapped transforms are equivalent to paraunitary filter banks.[26] Therefore, we can always regard any discrete, real, and orthogonal (lapped or block) transform as a filter bank.[26,32,33] The analysis filters' coefficients are the time-reversed basis functions elements.[26,32] Suppose the M basis functions have elements $p_k(n)$ ($k = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, L-1$). The equivalent analysis filter bank is shown in Fig. 3, where each filter [with coefficient $f_k(n)$] is equal to a basis function of the LOT, i.e., $f_k(n) = p_k(L-1-n)$ for $n = 0, 1, \ldots, L-1$. For the particular case of the LOT of M bands, $L = 2M$, but for the DCT we have $L = M$ (as any block transform). In Fig. 3, with $x(n)$ as the input signal to the filter bank,

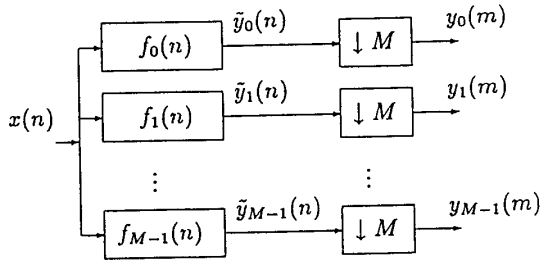


Fig. 3 Analysis section of a critically decimated M-band filter bank where $x(n)$ is the input signal and $\tilde{y}_k(mM)$ are the subband signals after filtering ($0 \le k \le M-1$). The subband signals are decimated resulting in $y_k(m) = \tilde{y}_k(mM)$. The filters' impulse responses $f_k(n)$ are the time-reversed basis functions of the transform.

$\tilde{y}_k(n)$ corresponds to each subband (filtered signals), and $y_k(n)$ is the subband signal after decimation. Let $F_k(e^{j\omega})$ be the frequency response of $f_k(n)$. Figure 4 shows the frequency response of the first three filters (basis functions) for a one-dimensional LOT with 8 bands (i.e., a 16×8 LOT matrix). Similar results for the DCT are found in Fig. 5. The same procedure can also be applied to nonuniform filter banks such as those resulting from the use of hierarchical structures. If, in Fig. 3, the input $x(n)$ has a power spectral density (psd) given by $S_x(\omega)$, and denoting the PSD of $\tilde{y}_k(n)$ and $y_k(n)$ as $S_{\tilde{y}_k}(\omega)$ and $S_{y_k}(\omega)$, we have:

$$S_{\tilde{y}_k}(\omega) = S_x(\omega) \left| F_k(e^{j\omega}) \right|^2 . \tag{12}$$

After the decimator, $y_k(n) = \tilde{y}_k(nM)$, and

$$S_{y_k}(\omega) = \sum_{r=0}^{M-1} S_{\tilde{y}_k}\left(\frac{\omega - 2\pi r}{M}\right) . \tag{13}$$

As

$$\int_a^b S_{\tilde{y}_k}(\omega)\, d\omega = \int_{2\pi-b}^{2\pi-a} S_{\tilde{y}_k}(\omega)\, d\omega ,$$

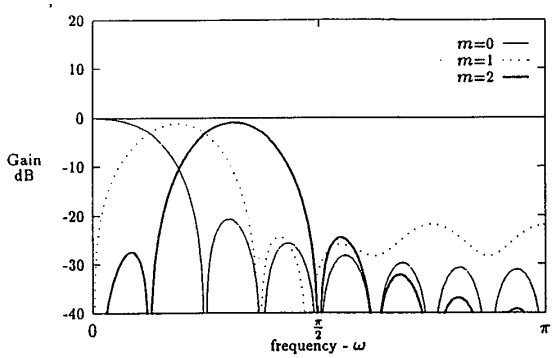


Fig. 4 Frequency response in decibels of the filters $f_m(n)$ corresponding to the first three basis functions of the LOT, i.e., $|F_m(e^{j\omega})|$, $m=0, 1, 2$.

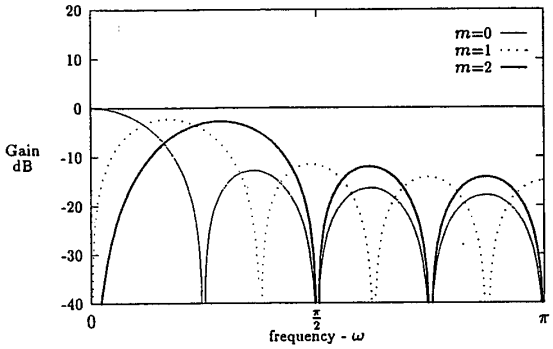


Fig. 5 Frequency response in decibels of the filters $f_m(n)$ corresponding to the first three basis functions of the DCT, i.e., $|F_m(e^{j\omega})|$, $m=0, 1, 2$.

the variance of y_k is given by

$$\zeta_k^2=\frac{1}{\pi}\int_0^{\pi} S_{y_k}(\omega)\,d\omega=\frac{1}{\pi}\int_0^{\pi} S_{\bar{y}_k}(\omega)\,d\omega \quad . \tag{14}$$

Alternatively, this result could be shown using the fact that if $u(n)$ is a stationary process, then $\text{var}[u(n)]=\text{var}[u(Mn)]$. Therefore, $\text{var}[\bar{y}_k(n)]=\text{var}[y_k(n)]$ and the preceding equation is also true.

Roughly, if a signal is filtered by $H_D(e^{j\omega})$, the signal and its filtered version would be indistinguishable for the observer to whom $H_D(e^{j\omega})$ is a perfect sensitivity model. If this signal has a flat PSD (white noise), the filtered signal has the PSD shaped by the filter, letting one know the relative importance of each frequency component for the observer. If this colored signal is split into subbands, as when using the LOT, how can we measure the importance of each subband component? A sampling in the frequency domain would be imprecise and very dependent on the phase of the sampling train, since there would be only M bands of width π/M. This bandwidth can be large enough to allow significant variations of the input PSD. Since we are measuring up to the second-order statistics in the image, and on those we may apply the weighting matrix, one possible

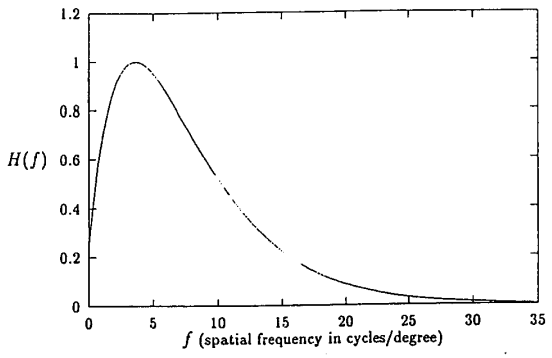


Fig. 6 HVS model function used in this paper.[2]

solution would be the measure of the variance of each band. These variances can provide the relative significance of each subband. Note that as M increases, L increases, and the filters are becoming close to ideal filters and the bandwidth is becoming narrower. In the limit, the approximations by sampling and by variance computation would yield the same results.

If a white noise with unit variance is input to the linear system $H_D(e^{j\omega})$, and its output is transformed using the LOT, then Eq. (14) is given by:

$$\zeta_k^2=\frac{1}{\pi}\int_0^{\pi} |H_D(e^{j\omega})|^2\,|F_k(e^{j\omega})|^2\,d\omega \tag{15}$$

The continuous HVS model function as used in Ref. 2 is plotted in Fig. 6. As previously stated, the frequency f is given in cycles per degree of visual angle subtended. The model is given by:

$$H(f)=2.46(0.1+0.25f)e^{-0.25f} \quad . \tag{16}$$

The corresponding weights ζ_k can be found using Eqs. (11) and (15).

The two-dimensional case is just an extension of these results, since the transform is separable. We are interested in weights ζ_{ij}, $(i,j)\in\Psi$, which can be derived from

$$\zeta_{ij}^2=\frac{1}{\pi^2}\int_0^{\pi}\int_0^{\pi} |H_D(e^{j\omega_1}, e^{j\omega_2})|^2 \times |F_{ij}(e^{j\omega_1}, e^{j\omega_2})|^2\,d\omega_1\,d\omega_2 \quad , \tag{17}$$

where

$$H_D(e^{j\omega_1}, e^{j\omega_2})=H_D(e^{j2\pi f_1}, e^{j2\pi f_2})=H(f_p/f_{\max}) \tag{18}$$

and

$$f_p=\sqrt{f_1^2+f_2^2}\ ; \quad |f_1|<f_{\max}\ , \quad |f_2|<f_{\max}$$

and

$$F_{ij}(e^{j\omega_1}, e^{j\omega_2})=F_i(e^{j\omega_1})\,F_j(e^{j\omega_2}) \quad . \tag{19}$$

In our application, we are weighting standard deviation values and we use ζ_{ij} instead of the squared value. Figure 7

0.6854	0.8698	0.9883	1.0000	0.9546	0.8703	0.7706	0.6793
0.8698	0.9371	0.9930	0.9821	0.9294	0.8457	0.7475	0.6598
0.9883	0.9930	0.9963	0.9606	0.8987	0.8154	0.7194	0.6362
1.0000	0.9821	0.9606	0.9114	0.8458	0.7659	0.6752	0.5984
0.9546	0.9294	0.8987	0.8458	0.7816	0.7073	0.6241	0.5543
0.8703	0.8457	0.8154	0.7659	0.7073	0.6409	0.5667	0.5047
0.7706	0.7475	0.7194	0.6752	0.6241	0.5667	0.5028	0.4493
0.6793	0.6598	0.6362	0.5984	0.5543	0.5047	0.4493	0.4024

(a) $\alpha = 4$; $f_{max} = 9$ cycles/degree

0.7460	0.9223	1.0000	0.9542	0.8566	0.7341	0.6071	0.5101
0.9223	0.9686	0.9836	0.9214	0.8222	0.7051	0.5829	0.4911
1.0000	0.9836	0.9515	0.8742	0.7749	0.6653	0.5503	0.4655
0.9542	0.9214	0.8742	0.7955	0.7032	0.6051	0.5021	0.4265
0.8566	0.8222	0.7749	0.7032	0.6222	0.5375	0.4483	0.3824
0.7341	0.7051	0.6653	0.6051	0.5375	0.4665	0.3916	0.3356
0.6071	0.5829	0.5503	0.5021	0.4483	0.3916	0.3312	0.2854
0.5101	0.4911	0.4655	0.4265	0.3824	0.3356	0.2854	0.2468

(b) $\alpha = 5$; $f_{max} = 11.2$ cycles/degree

0.8090	0.9702	1.0000	0.8988	0.7576	0.6112	0.4710	0.3804
0.9702	0.9920	0.9627	0.8533	0.7171	0.5803	0.4476	0.3630
1.0000	0.9627	0.8966	0.7846	0.6583	0.5354	0.4145	0.3379
0.8988	0.8533	0.7846	0.6845	0.5759	0.4714	0.3676	0.3012
0.7576	0.7171	0.6583	0.5759	0.4877	0.4024	0.3169	0.2610
0.6112	0.5803	0.5354	0.4714	0.4024	0.3348	0.2664	0.2206
0.4710	0.4476	0.4145	0.3676	0.3169	0.2664	0.2146	0.1789
0.3804	0.3630	0.3379	0.3012	0.2610	0.2206	0.1789	0.1498

(c) $\alpha = 6$; $f_{max} = 13.4$ cycles/degree

0.8629	1.0000	0.9750	0.8228	0.6487	0.4928	0.3515	0.2769
1.0000	0.9933	0.9177	0.7674	0.6051	0.4622	0.3305	0.2616
0.9750	0.9177	0.8206	0.6824	0.5402	0.4162	0.2998	0.2384
0.8228	0.7674	0.6824	0.5696	0.4549	0.3541	0.2582	0.2060
0.6487	0.6051	0.5402	0.4549	0.3678	0.2897	0.2144	0.1717
0.4928	0.4622	0.4162	0.3541	0.2897	0.2307	0.1733	0.1394
0.3515	0.3305	0.2998	0.2582	0.2144	0.1733	0.1326	0.1073
0.2769	0.2616	0.2384	0.2060	0.1717	0.1394	0.1073	0.0872

(d) $\alpha = 7$; $f_{max} = 15.7$ cycles/degree

Fig. 7 Two-dimensional HVS weighting matrices for the LOT, assuming 256 pels in a line and blocks of 8×8 pels. The relative distance α and maximum frequency f_{max} are indicated.

0.8945	1.0000	0.9209	0.7295	0.5375	0.3865	0.2539	0.2005
1.0000	0.9644	0.8474	0.6684	0.4942	0.3581	0.2362	0.1872
0.9209	0.8474	0.7270	0.5746	0.4289	0.3143	0.2098	0.1663
0.7295	0.6684	0.5746	0.4589	0.3477	0.2581	0.1754	0.1387
0.5375	0.4942	0.4289	0.3477	0.2683	0.2022	0.1404	0.1108
0.3865	0.3581	0.3143	0.2581	0.2022	0.1543	0.1092	0.0862
0.2539	0.2362	0.2098	0.1754	0.1404	0.1092	0.0792	0.0627
0.2005	0.1872	0.1663	0.1387	0.1108	0.0862	0.0627	0.0499

(a) $\alpha = 4$; $f_{max} = 18$ cycles/degree

0.9608	1.0000	0.8236	0.5781	0.3739	0.2485	0.1360	0.1222
1.0000	0.9107	0.7255	0.5121	0.3343	0.2242	0.1239	0.1104
0.8236	0.7255	0.5746	0.4122	0.2747	0.1863	0.1057	0.0915
0.5781	0.5121	0.4122	0.3025	0.2071	0.1423	0.0835	0.0698
0.3739	0.3343	0.2747	0.2071	0.1460	0.1021	0.0622	0.0504
0.2485	0.2242	0.1863	0.1423	0.1021	0.0723	0.0452	0.0361
0.1360	0.1239	0.1057	0.0835	0.0622	0.0452	0.0295	0.0231
0.1222	0.1104	0.0915	0.0698	0.0504	0.0361	0.0231	0.0185

(b) $\alpha = 5$; $f_{max} = 22.4$ cycles/degree

1.0000	0.9676	0.7115	0.4434	0.2512	0.1646	0.0707	0.0878
0.9676	0.8317	0.6001	0.3796	0.2184	0.1433	0.0631	0.0754
0.7115	0.6001	0.4384	0.2857	0.1699	0.1111	0.0516	0.0568
0.4434	0.3796	0.2857	0.1928	0.1191	0.0779	0.0385	0.0387
0.2512	0.2184	0.1699	0.1191	0.0767	0.0507	0.0266	0.0245
0.1646	0.1433	0.1111	0.0779	0.0507	0.0338	0.0182	0.0164
0.0707	0.0631	0.0516	0.0385	0.0266	0.0182	0.0106	0.0087
0.0878	0.0754	0.0568	0.0387	0.0245	0.0164	0.0087	0.0082

(c) $\alpha = 6$; $f_{max} = 26.8$ cycles/degree

1.0000	0.8965	0.5850	0.3231	0.1591	0.1141	0.0347	0.0692
0.8965	0.7254	0.4715	0.2669	0.1343	0.0943	0.0303	0.0561
0.5850	0.4715	0.3165	0.1870	0.0985	0.0663	0.0236	0.0379
0.3231	0.2669	0.1870	0.1157	0.0641	0.0418	0.0165	0.0227
0.1591	0.1343	0.0985	0.0641	0.0375	0.0241	0.0106	0.0123
0.1141	0.0943	0.0663	0.0418	0.0241	0.0159	0.0069	0.0083
0.0347	0.0303	0.0236	0.0165	0.0106	0.0069	0.0035	0.0033
0.0692	0.0561	0.0379	0.0227	0.0123	0.0083	0.0033	0.0046

(d) $\alpha = 7$; $f_{max} = 31.4$ cycles/degree

Fig. 8 Two-dimensional HVS weighting matrices for the LOT, assuming 512 pels in a line and blocks of 8×8 pels. The relative distance α and maximum frequency f_{max} are indicated.

shows weighting matrices containing normalized ζ_{ij} for f_{max} as 9.0, 11.2, 13.4, and 15.7 cycles/degree. They represent $\alpha = 4, 5, 6, 7$, respectively, for $N = 256$. Values for α of 6 or 7 are more representative for broadcast TV viewing. Values of 4 or 5 fit modern PIT needs very well and approximate the situation in which a 256×256 pel image is displayed on the 640×480 resolution mode on a regular home PC monitor, with the observer in front of it, working on the computer. The same procedure is repeated for the matrices in Fig. 8, assuming $N = 512$. For this value of N and the same values of α, the maximum frequencies are 18.0, 22.4, 26.8, and 31.4 cycles per degree.

4 Implementation and Results

A 256×256 pel monochrome image is divided into 8×8 nonoverlapping blocks ($M = 8$) and the LOT is applied to each block. Based on the ac energies, the 8×8 blocks are grouped into eight different equally populated classes ($N_C = 8$). Thus, there are 32×32 blocks in the image ($N_B = 1024$). The dc coefficients are quantized with a uniform 7-bit quantizer, and B_{max} is set to 7. Therefore, the overhead in Eq. (7) is, as previously computed, 4552 bits and the amount of bits needed to code the dc coefficients is 7168. This yields a total of 11,720 bits sent prior to the transmission of the ac coefficients (approximately 0.18 bits/pel). The block classification map for the 256×256 monochrome "Lena" image is shown in Fig. 9. Classes 1 through 8 represent increasing energies of 2-D LOT blocks. Figure 10 shows maps with standard deviations. Classes 1, 3, 6, and 8 are chosen as examples, and the dc coefficient is not computed. The resulting bit-allocation map for the eight classes is presented in Fig. 11. Using these maps and the weighting matrix of Fig. 7 (for $\alpha = 6$), by means of Eq. (9) we get the order for the transmission of the ac coefficients as shown in Fig. 12.

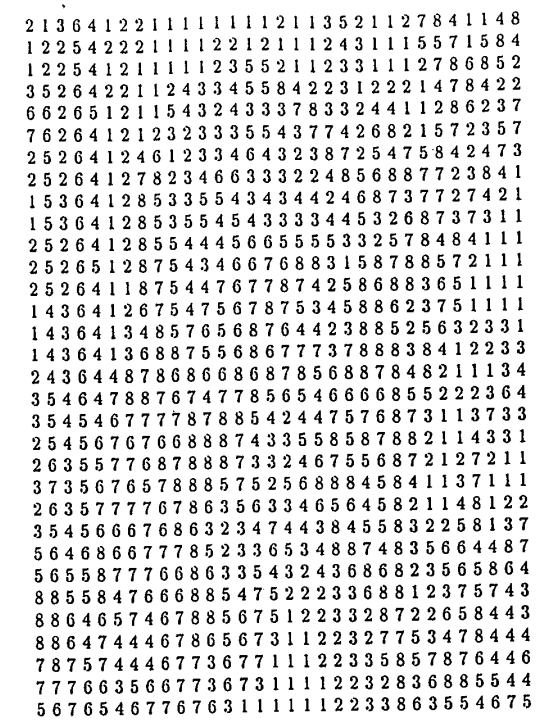


Fig. 9 Equally populated 32×32 classification map for the "Lena" image. Classes 1 through 8 represent increasing energies of 8×8 LOT blocks.

Class 1

0.0	8.1	3.9	4.0	2.6	2.9	2.2	2.3
7.7	4.5	3.3	3.0	2.1	2.3	2.1	2.0
3.9	2.6	2.6	2.4	2.1	1.8	1.7	1.8
2.7	2.1	2.3	2.0	2.1	1.7	1.8	1.8
2.1	2.0	1.7	1.7	1.7	1.6	1.7	1.6
2.1	2.0	1.8	1.7	1.9	1.7	1.5	1.6
1.8	1.8	1.7	1.5	1.6	1.6	1.4	1.6
1.9	1.8	1.7	1.6	1.6	1.7	1.5	1.7

Class 3

0.0	33.0	13.6	14.6	5.6	7.6	4.7	3.8
23.3	16.3	11.3	8.3	4.9	3.8	3.5	2.8
10.3	10.5	9.9	5.6	3.9	4.0	3.2	3.4
8.7	8.7	7.6	5.1	3.5	3.3	3.2	2.6
4.0	5.3	5.6	5.2	3.5	3.2	3.2	3.0
4.8	4.5	4.1	4.5	4.3	3.3	3.0	2.4
2.9	2.9	3.4	3.5	3.5	3.6	3.5	3.1
2.8	2.5	2.9	3.4	3.7	4.1	3.7	2.9

Class 6

0.0	121.1	54.8	39.1	22.9	24.1	13.8	12.8
67.3	51.8	42.1	27.4	20.4	17.8	14.0	12.3
29.7	31.1	29.8	25.8	21.4	15.7	15.0	12.1
22.5	20.7	22.1	19.7	16.8	11.3	10.0	7.8
12.9	13.9	12.3	14.0	11.4	12.4	9.8	8.2
12.1	11.4	11.6	8.2	9.4	9.6	6.5	6.6
5.6	5.8	5.8	6.2	5.8	6.2	5.1	5.9
6.2	5.7	5.5	5.3	6.4	5.1	4.4	4.9

Class 8

0.0	316.5	116.1	68.0	41.6	34.0	20.1	20.5
160.7	149.2	81.6	46.3	39.1	32.5	18.8	19.2
47.9	62.2	54.9	45.7	26.6	22.7	18.9	16.8
31.7	31.3	34.3	27.3	27.0	22.6	13.4	13.1
18.1	18.5	21.7	21.3	18.1	15.8	14.2	11.6
13.5	16.8	14.9	11.8	15.5	11.4	9.2	9.4
8.5	10.2	8.4	9.1	8.4	8.9	7.1	7.6
9.1	11.2	9.0	7.5	8.3	7.2	6.2	6.8

Fig. 10 Map with standard deviations of LOT coefficients in each class. Classes 1, 3, 6, and 8 are chosen as examples. The standard deviation for the dc coefficient is not shown.

The ac coefficients are well modeled by a Laplacian probability density function (pdf), but the blocks are classified according to their ac activities. If u is the amplitude of an ac coefficient, the actual important function is no longer its density function $p_U(u)$, but one conditional to the estimated standard deviation $p_U(u|\hat{\sigma})$. If there is just one class ($N_C = 1$), the Laplacian model fits well. At the other extreme, suppose there are as many blocks as classes (the overhead would be enormous). The variances would be computed from one element and would determine its amplitude completely. Therefore, the density would be an impulse. In this extreme case, all quantizers should only have two levels to indicate the sign of the coefficient. As long as we have few classes, these extreme cases do not apply. However, the lowest frequency ac coefficients (which have great influence in the classification process because they are larger) are well apart from having a Laplacian conditional density. As an example for a particular coefficient, suppose its standard deviation is estimated to be very large. This indicates that the coefficients on that coordinate $(m,n) \in \Psi$ belonging to the same class are expected to have high amplitudes, not amplitudes close to zero as in the Laplacian model. Generally, these large coefficients have low frequencies and have large numbers of bits allocated. Coefficients with one or two bits allocated generally do not have a great influence on the ac energy and are very close to the Laplacian model. In our constant distortion rule for bit allocation, we assumed that all the quantizers were optimized using the same pdf. Therefore, we have chosen the Gaussian density as the density model for our Lloyd-Max quantizers due to its greater robustness against pdf mismatches. Tests carried out (for 8 and 16 classes) using two sets of quantizers (for Laplacian and Gaussian pdfs), showed better performance for the Gaussian set of quantizers.

The reestimated standard deviations assume an integer number of bits allocated to each coefficient; hence, if we assume that all quantizer levels may be used, the quantizer should be a midrise one. For one- and two-bit quantizers optimized for a Gaussian input pdf, the inner reconstruction levels (positive or negative) are $0.798\sigma_1$ and $0.453\sigma_2$, respectively, where $\sigma_1 = c$ and $\sigma_2 = 2c$ represent the estimated standard deviations for those coefficients that have been allocated 1 and 2 bits, respectively. It is possible that some null or insignificant coefficients would have to be quantized using relatively high standard deviation values, and must be reconstructed as a nonzero component with a magnitude comparable to the standard deviation. In these cases, nonexistent frequency components emerge, resulting in annoying effects. For this reason, we decided to apply midtread quantizers with three levels and variable length coding, instead of quantizing with two or four levels. The standard

Class 1

7	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Class 2

7	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Class 3

7	3	1	2	0	1	0	0
2	2	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Class 4

7	3	2	2	1	1	0	0
2	3	2	1	1	0	0	0
1	1	2	1	1	0	0	0
1	1	1	1	0	0	0	0
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Class 5

7	4	3	3	2	2	1	1
3	3	2	2	1	1	0	0
2	2	2	1	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Class 6

7	5	3	3	2	2	1	1
4	3	3	2	2	2	1	1
3	3	3	2	2	2	2	1
2	2	2	2	2	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Class 7

7	5	4	3	3	2	2	2
4	4	4	3	3	2	2	2
3	3	3	3	2	2	1	1
2	2	2	2	2	2	2	1
1	2	2	2	2	1	1	1
1	2	1	1	1	1	1	1
0	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0

Class 8

7	6	4	4	3	3	2	2
5	5	4	3	3	3	2	2
3	4	3	3	2	2	2	2
3	3	3	2	2	2	1	1
2	2	2	2	2	2	1	1
1	2	2	1	2	1	1	1
1	1	1	1	1	1	0	1
1	1	1	1	1	0	0	0

Fig. 11 Bit-allocation maps for each class for the "Lena" image. The rate is 1 bit/pel including overhead. This is also the final stage of the PIT.

Class 1

–	137	–	–	–	–	–	–
138	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Class 2

–	70	114	119	–	–	–	–
139	124	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Class 3

–	40	115	56	–	161	–	–
71	61	111	125	–	–	–	–
116	112	118	–	–	–	–	–
120	126	131	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Class 4

–	41	52	57	140	162	–	–
72	31	49	127	144	–	–	–
117	113	54	132	155	–	–	–
121	128	133	151	–	–	–	–
–	145	156	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Class 5

–	13	19	27	73	87	181	198
42	32	50	62	146	170	–	–
53	51	55	134	157	172	–	–
58	63	135	152	167	–	–	–
141	147	158	168	175	–	–	–
163	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Class 6

–	3	20	28	74	88	182	199
14	33	16	64	76	95	184	202
21	17	24	67	82	99	122	213
59	65	68	79	90	177	194	217
142	148	159	169	176	190	208	225
164	171	173	178	191	205	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Class 7

–	4	9	29	43	89	107	149
15	12	6	34	45	96	109	153
22	18	25	37	83	100	187	214
60	66	69	80	91	105	136	218
143	77	84	92	103	192	209	226
165	97	174	179	193	206	222	229
–	185	188	195	210	–	–	–
200	203	215	219	–	–	–	–

Class 8

–	1	10	11	44	47	108	150
5	2	7	35	46	48	110	154
23	8	26	38	85	101	123	160
30	36	39	81	93	106	196	220
75	78	86	94	104	129	211	227
166	98	102	180	130	207	223	230
183	186	189	197	212	224	–	231
201	204	216	221	228	–	–	–

Fig. 12 Transmission order of the LOT coefficients among all classes for the "Lena" image. This order is found using the bit-allocation maps in Fig. 11, weighted by the HVS matrix in Fig. 7(c), according to Eq. (9). The transmission priority rules were defined in Sec. 2 based on these weighted matrices.

Fig. 13 Partially reconstructed images: (a) DCT 0.2 bit/pel, (b) LOT 0.2 bit/pel, (c) DCT 0.3 bit/pel, and (d) LOT 0.3 bit/pel.

deviations for quantization and reconstruction of these coefficients would remain the same, but the distortion rule and the average bit rate would be affected. However, the distortion increase, a result of going from four to three levels in the 2-bit quantizer, is partially compensated by the distortion decrease in going from two to three levels for the 1-bit quantizer. The same occurs with the bit-rate changes. In our simulations, both schemes yielded roughly the same bit rates, with the three-level scheme leading to images with higher signal-to-noise ratios (SNRs).

The HVS-weighted PIT described previously is extended to the 2-D DCT. The weighting matrix was found using the method described in Ref. 2 for $f_{max} = 13.4$ ($\alpha = 6$). Reconstructed images based on both LOT and DCT for several stages are shown in Fig. 13. Critical observation of these images indicates the improved fidelity and absence of block structure during the initial stages when LOT is used. In Table 1, a comparison of both methods is carried out, evaluating the SNR of reconstructed images at several stages for the "Lena" and "Girl" images. Since the HVS weighting is used only for prioritizing the transmission of coefficients, the SNR measure did not incorporate subjective weighting factors. If $u(m,n)$ and $\hat{u}(m,n)$ represent the original and reconstructed image, then the SNR is given by

$$\mathrm{SNR} = 10 \log_{10} \left\{ \frac{\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u^2(m,n)}{\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} [u(m,n) - \hat{u}(m,n)]^2} \right\} .$$

(e)

(f)

(g)

(h)

Fig. 13 (continued) Partially reconstructed images: (e) DCT 0.4 bit/pel, (f) LOT 0.4 bit/pel, (g) DCT 1.0 bit/pel, and (h) LOT 1.0 bit/pel.

5 Conclusions

A PIT scheme that incorporates adaptive classification in the transform domain and bit allocation based on the rate-distortion theory is presented. A general technique for developing HVS weighting of the transform coefficients is developed. Based on this, HVS weighting matrices applicable to LOT are obtained. The order in which the transform coefficients are transmitted is based on the estimated variances of these coefficients weighted by the human visual system sensitivity, measured in the 2-D LOT domain. Because these variances can be estimated at the receiver, overhead is limited to bit-allocation maps of the classes to which the blocks are grouped and to the classification of the blocks. The transform coefficients for all the classes during each stage are transmitted progressively such that a specified bit rate is reached for each stage. Visual comparison of the reconstructed images based on the LOT and DCT shows that the former yields subjectively superior images compared to the DCT in all stages.

Table 1 SNR (in decibels) resulting from intermediary reconstructed images at several bit rates for the "Lena" and "Girl" images.

Rate(bpp)	SNR			
	LOT	DCT	LOT	DCT
	LENA		GIRL	
0.2	16.10	15.18	17.22	16.21
0.3	19.43	18.41	20.27	19.41
0.4	21.00	20.55	22.68	21.93
0.5	22.74	22.39	24.30	23.79
0.6	23.68	23.29	25.21	24.98
0.8	25.35	25.15	26.98	26.76
1.0	26.81	26.67	28.50	28.27

Acknowledgment

This work was supported in part by Conselho Nacional de Desenvolvimento Cientifico e Tecnologico, CNPq, Brazil, under grant 200.804/90-1.

References

1. K. H. Tzou, "Progressive image transmission: a review and comparison of techniques," *Opt. Eng.* **26,** 581–589 (July 1987).
2. B. Chitprasert and K. R. Rao, "Human visual weighted progressive image transmission," *IEEE Trans. Commun.*, **COM-38,** 1040–1044 (July 1990).
3. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages and Applications,* Academic Press, San Diego (1990).
4. M. Rabbani and P. W. Jones, *Digital Image Compression Techniques,* SPIE Optical Engineering Press, Bellingham, WA (1991).
5. K. H. Tzou and S. E. Elnahas, "An optimal progressive transmission and reconstruction scheme for transformed images," ICC 86, pp. 413–418, Toronto, Canada (June 1986).
6. W. Gong, K. R. Rao, and M. T. Manry, "Progressive image transmission using Kohonen self organizing feature map," *IEEE Trans. Circuits Sys. Video Technol.* (under review).
7. E. R. Hamilton, "The JPEG standard for still picture coding," Society for Information Display (SID), 1991 International Symposium, Anaheim, CA, May 6–10, 1991.
8. A. Leger, T. Omachi, and E. K. Wallace, "JPEG still picture compression algorithm," *Opt. Eng.* **30,** 947–954 (July 1991).
9. W. Gong, K. R. Rao, and M. T. Manry, "Progressive image transmission using a self-supervised back-propagation neural network," *J. Electronic Imaging,* **1,** 88–94 (Jan. 1992).
10. S. E. Elnahas et al., "Progressive transmission of digital diagnostic images," *Appl. Digital Image Processing VIII,* Proc. SPIE **575,** 48–55 (Aug. 1985).
11. W. H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.* **COM-32,** pp. 225–232 (March 1984).
12. W. H. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, **COM-25,** 1285–1292 (Nov. 1977).
13. K. N. Ngan, "Adaptive transform coding of video signals," *IEE Proc.*, **129,** Pt. F, pp. 28–40 (Feb. 1982).
14. T. Saito, H. Takeo, K. Aizawa, H. Harashima, and H. Miyakawa, "Adaptive discrete cosine transform image coding using gain/shape vector quantization," *Proc. ICASSP '86,* pp. 129–132 (April 1986).
15. Y. S. Ho and A. Gersho,"Classified transform coding of images using vector quantization," *Proc. ICASSP '89,* 1890–1893 (April 1989).
16. J. Y. Nam and K. R. Rao, "Image coding using a classified DCT/VQ based on two channel conjugate vector quantization," *IEEE Trans. Circuits Sys. Video Technol.* **1,** 327–336 (Dec. 1991).
17. K. H. Tzou, T. R. Hsing, and J. G. Dunham, "Applications of physiological human visual system model to image compression," *Proc. SPIE* **504,** 419–424 (1984).
18. S. Ericsson, "Frequency weighted interframe hybrid coding," Rep. TRITA-TTT-8401, Telecommun. Theory, Royal Institute of Technology, Stockholm, Sweden (Jan. 1984).
19. H. Lohscheller, "A subjectively adapted image communication system," *IEEE Trans. Commun.* **COM-32,** 1316–1322 (Dec. 1984).
20. N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.* **COM-33,** 551–557 (June 1985).
21. K. N. Ngan, K. S. Leong, and H. Singh, "Cosine transform coding incorporating human visual system model," *Proc. SPIE* **707,** 165–171 (Sep. 1986).
22. P. M. Cassereau, D. H. Staelin, and G. De Jager, "Encoding of images based on a lapped orthogonal transform," *IEEE Trans. Commun.* **COM-37,** 189–193 (Feb. 1989).
23. H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Proc.* **ASSP-37,** 553–559 (April 1989).
24. H. S. Malvar, "Reduction of blocking effects in image coding with a lapped orthogonal transform," *Proc. ICASSP '88,* 781–784 (April 1988).
25. H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Proc.* **ASSP-38,** 969–978 (June 1990).
26. H. S. Malvar, *Signal Processing with Lapped Transforms,* Artech House, Norwood, MA (1992).
27. A. N. Akansu and F. E. Wadas, "On lapped orthogonal transforms," *IEEE Trans. Signal Proc.* **40,** 439–443 (Feb. 1992).
28. W. E. Lynch and A. R. Reibman, "The lapped orthogonal transform for motion-compensated video compression," *Proc. SPIE* **1605,** 285–296 (1992).
29. H. Jozawa and H. Watanabe, "Compatible coding by lapped orthogonal transform," presented at ITEC 90, 1990 ITE Annual Convention.
30. H. Jozawa and H. Watanabe, "Intrafield/Interfield adaptive lapped transform for compatible HDTV coding," presented at 4th International Workshop on HDTV and Beyond, Torino, Italy, Sep. 4–6, 1991.
31. S. Venkatraman, J. Y. Nam, and K. R. Rao, "Classified transform vector quantization of images using the lapped orthogonal transform," *Proc. SPIE* (in press).
32. M. Vetterli and D. Le Gall, "Perfect reconstruction filter banks: some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Proc.* **ASSP-37,** 1057–1071 (July 1989).
33. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing,* Prentice-Hall, Englewood Cliffs, NJ (1983).

Ricardo L. de Queiroz received the BSc degree from Universidade de Brasilia in 1987 and the MSc degree from Universidade Estadual de Campinas, Brazil, in 1990, both in electrical engineering. In 1990–1991, he was with the DSP research group at Universidade de Brasilia as a research associate. He is currently enrolled in the PhD program at University of Texas at Arlington. His main interests are multirate signal processing, filter banks, image and audio compression, and image databases. He is a member of IEEE and the Brazilian Telecommunications Society.

K. R. Rao received his BE degree from the University of Madras in 1952, MSEE and MSNE degrees from the University of Florida, Gainesville, in 1959 and 1960, respectively, and the PhD degree in electrical engineering from the University of New Mexico, Albuquerque, in 1966. Since 1966, he has been with the University of Texas at Arlington (UTA) where he is currently a professor of electrical engineering. He has published extensively in reviewed technical journals in the areas of discrete orthogonal transforms and digital image coding. He, along with two other researchers, introduced the discrete cosine transform in 1975, which has since become very popular in digital signal processing. He has organized and conducted short courses and conferences on thermoelectric energy conversion from 1969–1992. He is the coauthor of the books *Orthogonal Transforms for Digital Signal Processing.* (Springer-Verlag, 1975), *Fast Transforms: Analyses and Applications* (Academic Press, 1982), and *Discrete Cosine Transform—Algorithms, Advantages, and Applications* (Academic Press, 1990), as well as coauthor or editor on other works.

Modulated Lapped Transforms in Image Coding

Ricardo L. de Queiroz[1] and K. R. Rao

Electrical Engineering Department
University of Texas at Arlington
Box 19016, Arlington, TX, 76019
e-mail: queiroz@eepost.uta.edu and eekrr521@utacnvx.uta.edu

Abstract

The class of modulated lapped transforms (MLT) with extended overlap is investigated in image coding. The finite-length-signals implementation using symmetric extensions is introduced and human visual sensitivity arrays are computed. Theorectical comparisons with other popular transforms are carried and simulations are made using intraframe coders. Emphasis is given in transmission over packet networks assuming high rate of data losses. The MLT with overlap factor 2 is shown to be superior in all our tests with bonus features such as greater robustness against block losses.

1 Introduction

While block transforms became very popular in the image coding field, the lapped orthogonal transform (LOT) [1, 2] arose as a promising competitor to transforms such as the discrete cosine transform (DCT) [3], which is the block transform used in most image and video coding algorithms [3]. The advantage of lapped transforms [4] resides on the length of their basis functions, providing improved spectral selection better filtering capabilities, and on the extreme reduction of the blocking artifacts commonly present in block transform coding at low bit-rates. Furthermore, the concept of lapped transforms was established and proven to be equivalent to the concept of paraunitary FIR uniform filter banks [4, 5]. Under this point of view, both the LOT and the DCT are considered as special choices of paraunitary filter banks [4, 5, 6]. Cosine modulated filter banks [5] allow perfect reconstruction (PR) in paraunitary analysis-synthesis systems, using a modulation of a low-pass prototype by a cosine train. By a proper choice of the phase of the modulating cosine, Malvar developed the modulated lapped transform (MLT) [7], which led to the so-called extended lapped transforms (ELT)[4, 8]. The ELT allows several overlapping factors, generating a family of PR cosine modulated filter banks. Both designations (MLT and ELT) are frequently applied to this class of filter banks[4]. Other cosine-modulation approaches have also been developed (see, for example, [5, 9] and references therein) and the most significant difference among them is the low-pass prototype choice and the phase of the cosine sequence.

As ELTs are maximally decimated FIR uniform filter banks, let M be the number of filters, which is the number of channels, the decimation factor of the subbands, and the block size. In the ELTs, the filters' length L is basicaly an even multiple of the block size M, as $L = 2KM$, where K is the overlap factor. The analysis filters $(f_m(n))$ are time-reversed versions of the synthesis filters $(g_m(n))$ as in any paraunitary filter bank (for $m = 0, 1 \ldots, M-1$ and $n = 0, 1, \ldots, L-1$). The MLT-ELT class is defined by [4, 8]

$$g_m(n) = f_m(L-1-n) = h(n)\sqrt{\frac{2}{M}}\cos\left[\left(m+\frac{1}{2}\right)\left(n+\frac{M+1}{2}\right)\frac{\pi}{M}\right] \tag{1}$$

for $m = 0, 1 \ldots, M-1$ and $n = 0, 1, \ldots, L-1$. $h(n)$ is a symmetric window modulating the cosine sequence and the impulse response of a low-pass prototype (with cutoff frequency at $\pi/2M$) which is translated in frequency to M different frequency slots in order to construct the uniform filter bank. We will mostly use ELT with $K = 2$, which will be designated as ELT-2, while ELT with other overlap factors will be referred as ELT-K. We assume row-column separable implementation of the transform. Therefore, one-dimensional analysis of the transform implementation is sufficient for two-dimensional applications.

[1]This work was supported in part by CNPq, Brazil, under Grant 200.804-90-1.

0-8194-1482-4/94/$6.00

HVS Coding

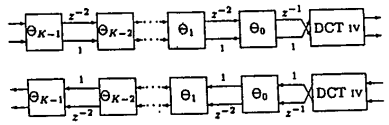


Figure 1. Flow graph for the direct (top) and inverse (bottom) ELT. Each branch carries $M/2$ samples.

2 Implementation algorithm

The ELT have as their major plus a fast implementation algorithm [4]. The algorithm is based on a factorization of the polyphase component matrix into a series of plane rotation stages and delays and a DCT type IV orthogonal transform in the last stage, which has fast implementation algorithms [3, 4]. The lattice-style algorithm [4] is shown in Fig. 1 for a ELT with generic overlap factor K. In Fig. 1 each branch carries $M/2$ samples, each Θ_i stage contain just $M/2$ orthogonal butterflies to implement $M/2$ plane rotations, and both analysis (forward transform) and synthesis (inverse transform) flow-graphs are shown. Let 0 and I be the null and identity matrices, respectively, of size $M/2 \times M/2$. Also, let J be the $M/2 \times M/2$ counter-identity matrix as:

$$\mathbf{J} = \begin{bmatrix} 00\cdots01 \\ 00\cdots10 \\ \vdots \quad \vdots \\ 01\cdots00 \\ 10\cdots00 \end{bmatrix}.$$

The stages Θ_n contain the plane rotations and are defined by

$$\Theta_n = \begin{bmatrix} -\mathbf{C}_n & \mathbf{S}_n\mathbf{J} \\ \mathbf{J}\mathbf{S}_n & \mathbf{J}\mathbf{C}_n\mathbf{J} \end{bmatrix}, \tag{2}$$

$$\mathbf{C}_n = diag\left\{\cos(\theta_{0,n}), \cdots, \cos(\theta_{\frac{M}{2}-1,n})\right\} \quad \mathbf{S}_n = diag\left\{\sin(\theta_{0,n}), \cdots, \sin(\theta_{\frac{M}{2}-1,n})\right\}$$

$\theta_{i,j}$ are the rotation angles and free parameters in the design of an ELT. We will use the optimized angles presented in [4]. The plane rotations define the window $h(n)$ and the DCT-IV generates the cosine terms in (1).

One problem to implement this algorithm resides on the transform applied to blocks near the borders of the image. As the transform contains overlap, samples outside the image boundaries may be included in the analysis section. On the other hand, extra transformed blocks are needed in the synthesis process to reconstruct the signal. Periodic extensions can solve this problem, at the expense of inserting artificial edges caused by unequal luminance levels in the extremes of the image. Symmetric extensions are desirable, because they do not include artificial edges and maintain polyphase normalization across the image boundaries. Malvar devised a solution for the finite-length-signals implementation of ELTs [4]. His solution envolves change of the filter bank near the boundaries and is back-to-back orthogonal. In [10, 11] non-orthogonal solutions are found using sample extensions and post-processing techniques and, in our tests, these solutions, using symmetric extension, proved to have better performance than the orthogonal method in [4]. Fig. 2(a) shows the flow-graph for PR implementation of the ELT-1 using symmetric extensions. In this figure,

$$\mathbf{Z}_A^{(1)} = (\mathbf{S}_0 - \mathbf{C}_0)\mathbf{J}, \tag{3}$$

$$\mathbf{Z}_B^{(1)} = \mathbf{J}(\mathbf{S}_0 + \mathbf{C}_0). \tag{4}$$

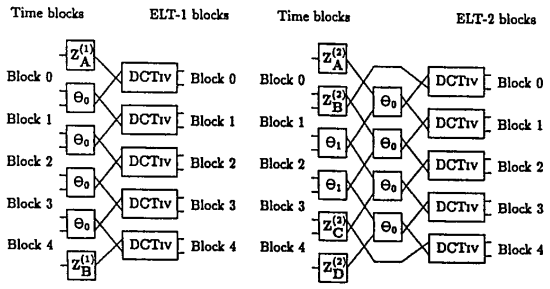


Figure 2. Flow graph for finite-length signals. (a) ELT-1 ($K=1$); (b) ELT-2 ($K=2$). Each branch carries $M/2$ samples. Forward transform is performed by following the flow-graph from left to the right, while inverse transform is performed by following the flow-graph in the opposite direction and substituting the Z matrices by their inverses.

Similarly, Fig. 2(b) shows the flow-graph for PR implementation of the ELT-2 using symmetric extensions, where

$$Z_A^{(2)} = (S_1 - C_1)J, \tag{5}$$

$$Z_B^{(2)} = \begin{bmatrix} -C_1 & S_1 J \\ (S_0 - C_0)S_1 & (S_0 + C_0)C_1 J \end{bmatrix}, \tag{6}$$

$$Z_C^{(2)} = \begin{bmatrix} J(C_0 - S_0)C_1 & J(C_0 + S_0)S_1 J \\ JS_1 & JC_1 J \end{bmatrix}, \tag{7}$$

$$Z_D^{(2)} = J(S_1 + C_1). \tag{8}$$

In these flow-graphs, each branch carries $M/2$ samples and analysis is accomplished by following the paths from left to the rigth, while the synthesis (inverse transform) is achieved by following the paths from right to the left, replacing the Z matrices by their inverses. Note that $Z_A^{(1)}, Z_B^{(1)}, Z_A^{(2)}$ and $Z_D^{(2)}$ are simple counter-diagonal matrices and their inverses have the same basic format. $Z_B^{(2)}$ and $Z_C^{(2)}$ are composed by $M/2$ butterflies, but the lattice is no longer orthogonal (see (6) and (7)). Their inverses are obtained by inverting each of the butterflies. As a result, both analysis or synthesis have the same fast algorithm. The DCT-IV and Θ_n matrices do not need replacement in synthesis because they are both symmetric and orthogonal.

These algorithms for ELT-1 and ELT-2 were found by using a symmetric extension and applying regular ELT flow-graph to the extended sequence. Then, it is found a size-limited flow-graph that would be equivalent. Values of K greater than 2 can also be found, but we will use mostly the ELT-2 and occasionaly the ELT-1.

Table I. G_{TC} in dB and implementation complexity (C) in FLOPS for various transforms and block sizes. For the tree structured filter banks, full-tree is applied and $M = 2^S$, where S is the number of stages of the tree.

	C	G_{TC}	C	G_{TC}	C	G_{TC}
	$M=4$		$M=8$		$M=16$	
DCT	3.5	7.57	5.25	8.83	7.13	9.46
LOT	7	7.95	9.50	9.20	11.75	9.69
ELT-1	7	8.12	9	9.33	11	9.84
ELT-2	9	8.39	11	9.48	13	9.90
QMF-8A	16	8.31	24	9.32	32	9.67
CQF-8A	18	8.44	27	9.47	36	9.84
QMF-16B	32	8.52	48	9.56	64	9.92
CQF-16B	34	8.54	51	9.58	68	9.94
IDEAL	∞	8.56	∞	9.59	∞	9.94

3 Theorectical comparisons with other transforms

The coding gain G_{TC}, in dB, of a transform/subband scheme is defined as [12]

$$G_{TC} = 10\log_{10}\left(\frac{\frac{1}{M}\sum_{i=0}^{M-1}\sigma_i^2}{\left(\prod_{i=0}^{M-1}\sigma_i^2\right)^{1/M}}\right), \tag{9}$$

with σ_i^2 as the variance of the i-th subband signal (transform coefficient) for $i = 0, \ldots, M-1$. Under certain assumptions, it measures the gain of transform coding over PCM coding [12], i.e., measures the gain in terms of signal-to-noise ratio (SNR) one can obtain by transforming the signal before coding it. Actually, it measures the average performance of the transform and its potential regarding compacting the most of the energy of the signal in fewer coefficients. Implementation complexity (C) will be measured here by the number of floating-point operations (additions plus multiplications) per sample (FLOPS) required to implement the one-dimensional transform.

Another way to generate transforms with longer overlap is based on the hierarchical connection of two-band filter banks following the paths of a binary tree. For parallel M-band systems, the full-tree is applied. If S stages of filter banks are cascaded, the resulting filter bank will have $M = 2^S$ channels and the resulting filters have length $(M-1)(L_{2B}-1)+1$, where L_{2B} is the length of the filters in the two-band filter bank used as a basic cell for the hierarchical structure.

In Table I are shown G_{TC} and C for the DCT, LOT, ELT-1, and ELT-2. Additionally, we included the same parameters for the tree-structured filter banks based on two-band systems with filters with 8 and 16 taps. We used Smith and Barnwell conjugate quadrature filters (CQF) [13] and Johnston's quadrature mirror filters (QMF) [14]. Note that 8-tap filters lead to equivalent filter banks whose filters have length closer to ELT-2. The "IDEAL" entry in this table refers to ideal brick-wall filters, which only can be implemented using infinite-length filters. The input signal was assumed to follow an AR(1) model with adjacent-sample correlation $\rho = 0.95$.

From Table I, we can see that ELT-2 has coding gain similar to 8A tree-structured filter banks at a much lower complexity. The complexity of the DCT is unbeatably low, and LOT and ELT-2 are regarded as improvements for which the trade-off between costs and benefits has to be taken into account. LOT has proven to be superior to DCT leading to more pleasant images even at high compression rates. We will show later that ELT-2 surpasses LOT performance with bonus features.

One of the incentives to study ELTs of larger overlap for image coding resides in their longer basis functions and, therefore, in their potential for better spectral selectivity of each subband filter. In fact, the ELT-2 has filters with good stopband attenuation. The filtering capability is supposed to be reflected by G_{TC} measurements, in the case of

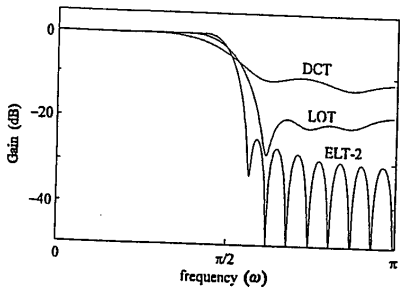


Figure 3. Frequency response of a low-pass filter with cut-off frequency = $\pi/2$ produced by the first four filters of a $M = 8$ transform. Plots for DCT, LOT, and ELT-2 are shown.

non-layered image coding. However, in compatible coding [15], filtering performance is a plus. In this approach, an image, let us say, of size $2N \times 2N$ is encoded using any transform/quantization method and the receiver makes the option of decoding the $2N \times 2N$ image or a reduced version of it, such as a $N \times N$ image. A straightforward method can be: inverse transform followed by anti-aliasing filtering and subsampling. As a faster and more efficient approach, one can retain $M/2 \times M/2$ low-frequency coefficients in each block (out of $M \times M$) and perform a pruned inverse transform, resulting in a reconstructed image at a lower spatial resolution. This is equivalent to tranform domain filtering followed by subsampling. Fig. 3 shows the frequency response of the first four filters for the 8-channel DCT, LOT, and ELT-2. They are combined into one band, which is the frequency response of the lowpass filter actually implemented when four (out of eight) coefficients are retained.

4 HVS weighting array

Our intention is restricted to the determination of a spatial response weighting matrix for use with the ELT-2 coefficients and we will use the process described in [17] to find a human visual sensitivity (HVS) weighting function for any transform. We will use a linear function to describe the HVS to spatial variations, because it was succesfully used in the past, although the HVS model response is not linear. Assume that the image area is square and the viewer is right in front of its geometrical center, at a distance α times longer than the image width. Assume, also, that the image contains N pels in a row or column and that the viewer can observe the same density of pels per spatial degree in any region of the screen. Given a linear transfer function representing the unidimensional spatial HVS as $H(f)$ (f given in cycles per degree of the visual angle subtended), the maximum visible frequency can be found as:

$$f_{max} = \frac{N}{4 \arctan\left(\frac{1}{2\alpha}\right)} \text{ cycles/degree.} \tag{10}$$

We can represent a discrete sensitivity function as $H_D(e^{j\omega}) = H_D(e^{j2\pi f}) = H(f/f_{max})$ for $|f| < f_{max}$. This representation will be accurate if $H(f) = 0$ for $|f| > f_{max}$. Let $F_k(e^{j\omega})$ be the frequency response of the k-th synthesis filter $f_k(n)$. The HVS weighting coefficients ζ_{ij} (for $0 \le i, j \le M-1$) are found from

$$\zeta_{ij}^2 = \frac{1}{\pi^2} \int_0^\pi \int_0^\pi |H_D(e^{j\omega_1}, e^{j\omega_2})|^2 \, |F_{ij}(e^{j\omega_1}, e^{j\omega_2})|^2 \, d\omega_1 \, d\omega_2, \tag{11}$$

(a)

0.6516	0.8667	0.9843	1.0000	0.9511	0.8669	0.7672	0.6629
0.8667	0.9402	0.9923	0.9839	0.9278	0.8432	0.7457	0.6445
0.9843	0.9923	0.9923	0.9588	0.8943	0.8097	0.7156	0.6190
1.0000	0.9839	0.9588	0.9112	0.8432	0.7613	0.6729	0.5829
0.9511	0.9278	0.8943	0.8432	0.7771	0.7010	0.6205	0.5387
0.8669	0.8432	0.8097	0.7613	0.7010	0.6331	0.5617	0.4892
0.7672	0.7457	0.7156	0.6729	0.6205	0.5617	0.5000	0.4372
0.6629	0.6445	0.6190	0.5829	0.5387	0.4892	0.4372	0.3838

(b)

0.7775	0.9744	0.9974	0.8985	0.7524	0.6018	0.4665	0.3539
0.9744	1.0000	0.9625	0.8539	0.7136	0.5716	0.4443	0.3380
0.9974	0.9625	0.8914	0.7811	0.6523	0.5246	0.4099	0.3135
0.8985	0.8539	0.7811	0.6821	0.5716	0.4628	0.3644	0.2808
0.7524	0.7136	0.6523	0.5716	0.4823	0.3939	0.3131	0.2435
0.6018	0.5716	0.5246	0.4628	0.3939	0.3250	0.2610	0.2051
0.4665	0.4443	0.4099	0.3644	0.3131	0.2610	0.2120	0.1683
0.3539	0.3380	0.3135	0.2808	0.2435	0.2051	0.1683	0.1351

(c)

0.8627	1.0000	0.9106	0.7199	0.5262	0.3665	0.2470	0.1634
1.0000	0.9632	0.8377	0.6593	0.4843	0.3395	0.2303	0.1531
0.9106	0.8377	0.7141	0.5635	0.4185	0.2970	0.2038	0.1369
0.7199	0.6593	0.5635	0.4499	0.3394	0.2449	0.1707	0.1163
0.5262	0.4843	0.4185	0.3394	0.2609	0.1919	0.1362	0.0943
0.3665	0.3395	0.2970	0.2449	0.1919	0.1439	0.1041	0.0734
0.2470	0.2303	0.2038	0.1707	0.1362	0.1041	0.0768	0.0551
0.1634	0.1531	0.1369	0.1163	0.0943	0.0734	0.0551	0.0402

(d)

1.0000	0.9838	0.7103	0.4354	0.2449	0.1309	0.0680	0.0354
0.9838	0.8350	0.5952	0.3712	0.2127	0.1155	0.0608	0.0320
0.7103	0.5952	0.4319	0.2784	0.1649	0.0921	0.0495	0.0264
0.4354	0.3712	0.2784	0.1868	0.1152	0.0667	0.0369	0.0201
0.2449	0.2127	0.1649	0.1152	0.0740	0.0445	0.0255	0.0142
0.1309	0.1155	0.0921	0.0667	0.0445	0.0278	0.0165	0.0095
0.0680	0.0608	0.0495	0.0369	0.0255	0.0165	0.0101	0.0060
0.0354	0.0320	0.0264	0.0201	0.0142	0.0095	0.0060	0.0036

Figure 4. Two-dimensional normalized HVS weighting arrays for the ELT-2 and $M = 8$. (a) $\alpha = 4$, $N = 256$, (b) $\alpha = 6$, $N = 256$, (c) $\alpha = 4$, $N = 512$, and (d) $\alpha = 6$, $N = 512$.

where

$$H_D(e^{j\omega_1}, e^{j\omega_2}) = H_D(e^{j2\pi f_1}, e^{j2\pi f_2}) = H(f_p/f_{max}), \tag{12}$$

where $f_p = \sqrt{f_1^2 + f_2^2}$ (for $|f_1| < f_{max}$ and $|f_2| < f_{max}$) and

$$F_{ij}(e^{j\omega_1}, e^{j\omega_2}) = F_i(e^{j\omega_1}) \, F_j(e^{j\omega_2}). \tag{13}$$

For the continuous one-dimensional HVS model given by [18]

$$H(f) = 2.46(0.1 + 0.25f) \, e^{-0.25f}, \tag{14}$$

we show in Fig. 4 normalized HVS weighting arrays of size 8×8 ($M = 8$) for the ELT-2, using several values of α and N. Note that the same f_{max} can accomodate different combinations of N and α. However, α in the range of 5 thru 6 is more common to displays in computer monitors, while the range of 4 thru 8 is more common to TV. For uniform quantization such as in JPEG baseline coder [19], the step size for each coefficient in a block can be c/ζ_{ij}, where c is a scaling constant.

5 Information loss in ATM networks

Asynchronous transfer mode (ATM) networks are gaining acceptance lately. The signal data is grouped into fixed size cells (packets) and transfered thru the network, and cells are sent when required and do not obey a fixed transmission rate. This allows integration of various services and more efficient channel sharing. Most protocols provide cell prioritization to protect more important data, as cell losses can occur. For transform coding, it is important to find ways to recover lost information with minimum possible distortion. We assume that as a single cell-loss occurs, all the information for a block is lost, except for the DC coefficient which is protected. LOT has proven to be robust against errors in ATM networks [16]. In [16], several recovery methods for the LOT were tested, including: (i) setting all AC coefficients to zero; (ii) coefficient averaging among neighboring blocks; (iii) inverse methods, with or without enhancement; (iv) least squares. We will limit ourselves to simple reconstruction by setting all AC coefficients to zero, which is the most most economical way to reconstruct the lost blocks. The ELT-2 is expected to perform better than LOT because of its larger overlapping. As the spatial region affected by the lost-block increases, the error is locally less intense. Fig. 5(a) shows the original 256 x 256 pels image Lena. We transformed this image using the DCT, LOT, and ELT-2, and deleted all the coefficients of a single block except for the DC term. After respective inverse transforms, Fig. 5(b) shows a zoom of the region where a lost-block occured, using the DCT. From this image, we can clearly see where the lost-block was located. Fig. 5(c) shows the same results for the LOT and Fig. 5(d) repeats the experiment for the ELT-2. We can see that ELT-2 performed fairly better than DCT and LOT when a cell loss occurs.

6 Image Coding Simulations

We have compared the ELT-1 and ELT-2, using two intraframe image coders, named JPEG baseline coder (JPEG) [19] and the improved Chen-Smith (ICS) coder [20]. The former is based on thresholding, while the latter outperforms JPEG and follows the zonal sampling philosophy. In both, the DCT was merely replaced by the other transforms. The SNR (not peak-SNR) results comparing ELT-2, ELT-1, LOT, and DCT are found in Table II, using both JPEG and the ICS coder. In these measurements the SNR was computed by skipping the first and last 4 samples of each column or row. This is to prevent the errors on the borders (using the ELTs) from affecting the SNR values, since they are generally invisible, occuring more intensively on the last pixel in each row-column [11] and being masked by the background.

Fig. 6 shows the 256 x 256 pels image Lena coded at 0.8 bits/pel (bpp) using both the JPEG algorithm and both the DCT and the ELT-2. In these we simulated 5% rate of lost blocks (51 blocks are lost).

7 Conclusions

The ELT-2 is proven to be very robust against cell losses due to its larger overlap. However, this greater overlap is achieved with only a small increase in computation. The finite-length implementations, presented here, allow the ELT-2 to be efficiently computed even near the borders and this transform reveals to be a very attractive alternative for image coding, replacing the DCT in applications such as still-frame image coding (JPEG) or in other coders.

References

[1] H. S. Malvar, "Reduction of blocking effects in image coding with a lapped orthogonal transform," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, Glasgow, Scotland, pp. 781-784, Apr. 1988.

[2] H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 37, pp. 553-559, Apr. 1989.

[3] K. R. Rao and P. Yip, *Discrete Cosine Transform : Algorithms, Advantages, Applications*. San Diego, CA : Academic Press, 1990.

[4] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech House, 1992.

Table II. SNR in dB of simulation results using the JPEG algorithm and comparing DCT, LOT, ELT-1 and ELT-2, for several bit-rates (in bpp).

	Lena 256 x 256				Jet 512 x 512			
Rate(bpp)	0.4	0.6	0.8	1.0	0.15	0.25	0.35	0.5
	JPEG coder							
DCT	22.09	24.16	25.74	27.13	21.32	25.93	28.50	30.62
ELT-1	22.46	24.66	26.22	27.52	21.65	26.74	28.96	30.82
LOT	22.50	24.47	26.02	27.32	21.66	26.77	29.10	31.11
ELT-2	22.72	24.77	26.25	27.58	22.68	27.52	29.41	31.45
	ICS coder							
DCT	22.91	25.38	27.32	28.95	20.36	26.45	28.91	31.46
LOT	23.16	25.53	27.40	28.86	21.09	27.02	29.33	31.65
ELT-2	23.56	25.95	27.70	29.17	21.80	27.52	29.85	32.03

[5] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[6] M. Vetterli and D. Le Gall, "Perfect reconstruction filter banks : some properties and factorizations", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 37, pp. 1057-1071, July 1989.

[7] H.S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol.38, pp. 969-978, Jun. 1990.

[8] H. S. Malvar, "Extended lapped transform: fast algorithms and applications," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, Toronto, Canada, pp. 1797-1800, 1991.

[9] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, Vol. 40, pp. 770-783, Apr. 1992.

[10] R. L. de Queiroz, "Subband processing of finite length signals without border distortions," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, San Francisco, CA, vol. IV, pp. 613-616, 1992.

[11] R. L. de Queiroz, "Perfect reconstruction subband processing of finite length signals," preprint.

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

[13] M. J. T. Smith and T. P. Barnwell III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp 434-441, June 1986.

[14] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, Denver, CO, pp. 291-294, 1980.

[15] H. Jozawa and H. Watanabe, "Intrafield/Interfield adaptive lapped transform for compatible HDTV coding," *4th International Workshop on HDTV and Beyond*, Torino, Italy, Sept. 4-6, 1991.

[16] P. Haskel and D. Messerschmidt, "Reconstructing lost video data in a lapped orthogonal transform based coder," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, Albuquerque, NM, pp. 1985-1988, 1990.

[17] R. L. de Queiroz and K. R. Rao, "HVS weighted progressive transmission of images using the LOT," *Journal of Electronic Imaging*, vol. 1, pp. 328-338, July 1992.

[18] B. Chitprasert and K.R. Rao, "Human visual weighted progressive image transmission," *IEEE Trans. Commun.*, vol. 38, pp. 1040-1044, July 1990.

[19] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*," New York, NY: Van Nostrand Reinhold, 1993.

[20] E. M. Rubino, H. S. Malvar and R. L. de Queiroz, "Improved Chen-Smith image coder," *Proc. of IEEE Intl. Symp. Circuits and Systems*, Chicago, IL, pp. 267-270, May 1993.

Figure 6. Image compression results for 256 x 256-pels image Lena. All compressed images are subject to a block-loss rate of 5%. (a) Original image at 8 bpp; (b) Image (a) compressed to 0.8 bpp using DCT and JPEG coder; (c) Image (a) compressed to 0.8 bpp using ELT-2 and JPEG coder;

Fig. 6(b)

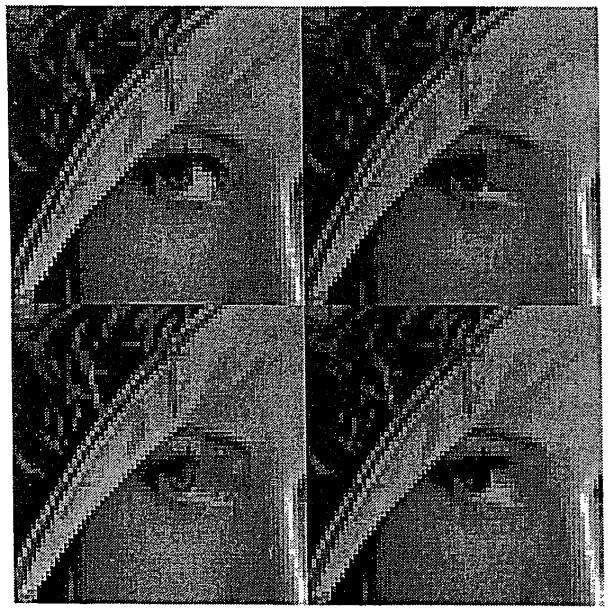


Figure 5. Trivial image reconstruction when all AC coefficents of a single block are lost. The AC coefficients are set to zero in this block. Top left corner, original image zoom. Top right corner, reconstructed image using DCT. Bottom left corner, reconstructed image using LOT, and on the bottom right corner, the same for the ELT-2.

Fig. 6(c)

Vol. 3

M11.2

A Perceptually Tuned Sub-band Image Coder With Image Dependent Quantization and Post-quantization Data Compression

Robert J. Safranek *James D. Johnston*
AT&T Bell Laboratories
Murray Hill, NJ

Abstract

In this paper we present a 16 band sub-band coder, arranged as 4 equal width sub-bands in each dimension, that uses an empirically derived perceptual masking model to set noise-level targets not only for each sub-band but also for each pixel in a given sub-band. The noise-level target is used to set the quantization levels in a DPCM quantizer. The output from the DPCM quantizer is then encoded, using an entropy-based coding scheme, in either 1x1, 1x2, or 2x2 pixel blocks. The type of encoding depends the statistics in each 4x4 sub-block of a particular sub-band. One set of codebooks, consisting of less than 100,000 entries, is used for all images, while the codebook subset used for any given image is dependent on the distribution of the quantizer outputs for that image. A block elimination algorithm takes advantage of the peaky spatial energy distribution of sub-bands to avoid using bits for quiescent parts of a given sub-band. Using this system, high quality output is obtainable at bitrates 0.1 to 0.9 bits/pixel, while nearly transparent quality requires 0.3 to 1.5 bits/pixel.

1. Introduction

In general, the current generation of low bitrate (< 1bpp) Black and White image coders provide a quality level of good to very good. Many applications, such as remote slideshows, would benefit from higher quality. To achieve this level of performance, we believe that knowledge of human visual perception should play a strong part in the coder design process. Our goal in this work, was to develop a visual perceptual quality metric which would provide nearly transparent quality to a coded image. In addition, this metric should be image independent. That is, it should perform equally well over a wide range of image input, say flat field to strong irregular texture, with no image specific tuning. This paper will present a system that uses this perceptual metric in conjunction with sub-band filtering, DPCM coding of sub-bands and multidimensional Huffman compression to provide nearly transparent coding of a wide variety of images at rates of less than 1 bit/pixel.

2. Sub-band Analysis

In order to exploit the generally lowpass characteristic of images, each image is first passed through a separable Generalized Quadrature Mirror Filter (GQMF) bank [Cox,Woods], after the mean of the image is calculated and removed. The mean is quantized to 8 bits (0-255) and retained for transmission to the decoder. Each of the 1-dimensional GQMF filters decompose the input image into 4 bandpass sub-images with one stage of filtering. This contrasts with the 2 stages required with conventional QMF filters. Since the filters are applied in both the horizontal and vertical dimensions, this results in 16 total sub-bands, numbered as shown below.

Sub-band Numbering			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Table 1 - The sub-bands are numbered using this scheme.

The GQMF filter that was used has a first sidelobe suppression of >48dB, which ensures perfect reconstruction of an 8 bit/pixel image (ignoring edge effects). A contrast enhanced example of the sub-band images, where the range in each sub-band is stretched to full scale, is shown here for a text image:

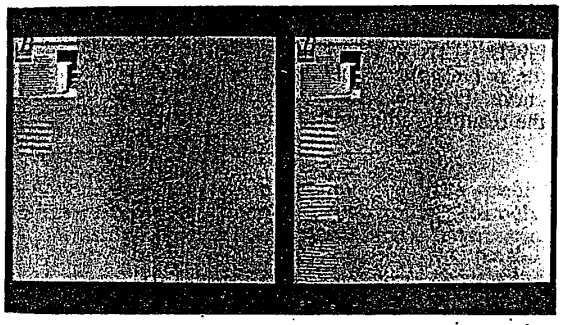


Figure 1: Here are sub-band images of grayscale text. The right image is contrast enhanced with each sub-band stretched to use the full gray scale range.

The actual mean energy for this image is 123 and the peak level in each sub-band is:

947	442	282	178
339	189	165	134
175	134	76	88
141	90	70	45

Table 2 - Presented here are the peak values in each sub-band for the text image of Figure 1.

3. Perceptual Masking Model

In the perceptual masking model, we use the local mean and variance to calculate a noise tolerance relative to the observed noise sensitivity of that sub-band given a uniform background grey level of 127.

CH2673-2/90/0000-1945 $1.00

3.1 Obtaining the Base Sensitivity

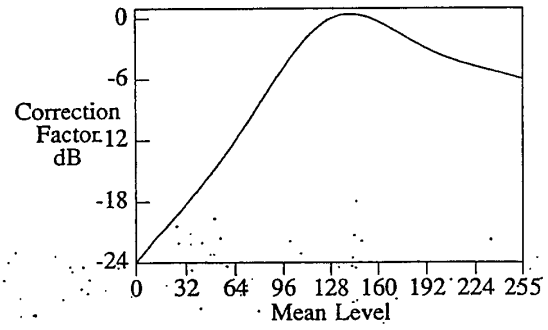
The base sensitivity for each sub-band was established in an informal test using 3 trained subjects. A set of 512x512 images with a constant grey level of 127 (on a scale of 0-255) were created, and uniformly distributed random noise of known energy was added to the center 64x64 pixels of each sub-band in turn. Taking into account the 4:1 decimation ratio, the reconstructed pictures have a 256x256 square of noise in the center. For each sub-band, the energy level of the noise was adjusted until the observers could not reliably determine if the reconstructed image did or did not contain the "noise square". The images were viewed in a darkened room on a Sun 3/110 workstation screen at 6 times the image height. The results of this sensitivity test were:

0.25	0.4	2.0	6.0
0.5	1.0	4.0	8.0
2.0	3.0	4.0	6.0
3.0	6.0	10.0	11.0

Table 3 - RMS noise sensitivity threshold for each sub-band. The order corresponds with Figure 1.

3.2 Sensitivity Adjustment for Brightness

The next step in determining the perceptual model was to vary the image grey scale background, and determine the change in sensitivity of band 0 for varied background grey levels. This test was run in the same manner as the previous test, yielding a brightness correction curve. For the specific conditions in this coder, the resulting adjustment curve is:



The brightness adjustment was spot-checked in other low frequency bands and found to predict the thresholds reasonably well. A better model could be obtained by running this correction test for each sub-band.

3.3 Texture Masking Adjustment

The base sensitivity and brightness adjustment provide a perceptual threshold which attempts to account for the human visual systems sensitivity to frequency content and image brightness *for a flat-field image*. Since humans are more sensitive to noise in flat-fields than in textured regions, this model provides a conservative perceptual threshold. Smooth image regions would be coded to an appropriate quality level, but textured regions would be greatly over-coded. Therefore, a texture masking adjustment was incorporated to the perceptual model.

The texture masking adjustment is a function of the "texture energy" at each image location. It is comprised of the weighted sum of the local (either 2x2 or 1x1 pixel, depending on the target quality) energy in each sub-band other than band zero plus the variance of band zero over the same locality (the variance is always taken over a 2x2 area with the target pixel in the upper left corner). The weights for each sub-band are determined empirically from the visual system's modulation transfer function [Cornsweet]. That is

$$TexEnergy(x,y) = \sum_{s=1}^{15} MTFweight(s)*Energy(s,x,y) + MTFweight(0)*variance((x,y),(x+1,y),(x,y+1),(x+1,y+1))$$

Where *TexEnergy* is the measure of texture energy, x and y horizontal and vertical pixel indices, *MTFweight* is the empirical weight from [Clarke, p. 271], and *variance* an operator that returns the variance of the enclosed pixels. This provides a crude measure of how much masking energy is visible in each sub-band. his texture energy is raised to the power 0.07, and the energy threshold multipled (or added in the dB domain) by the texture component.

The final form of the perceptual threshold is

$$pt(s,x,y) = Base(s) - .15*\log_e(TexEnergy(x,y)) - BrightWeight*BrightCorr(x,y)$$

where x and y are pixel locations in a sub-band, s is the sub-band number, Base() is the base noise sensitivity from Table 3 (in dB), and *pt* is expressed as a PSNR. Shown below is a representation of the relative perceptual threshold function for the text image. Portions of the image that have large tolerance to coding errors are represented by dark pixels, while sensitive areas are indicated by light pixels.

4. DPCM Coding of Sub-bands

Each sub-band is coded using a DPCM coder with a variable uniform mid-riser quantizer. It uses a three point predictor optimized for each sub-band. The predictor coefficients are quantized to 5 bit accuracy and sent as side information. The quantizer step size is adjusted to ensure that the perceptual criterion is just met at most critical point in the sub-band. This ensures that every point in the sub-band receives a sufficiently high level of coding without overcoding the most sensitive position. Due to the wide dynamic range of the perceptual threshold values, adaptation of the quantizer step size will be advantageous. However, we have just begun testing a modified step-size algorithm that responds within each sub-band to the image texture information.

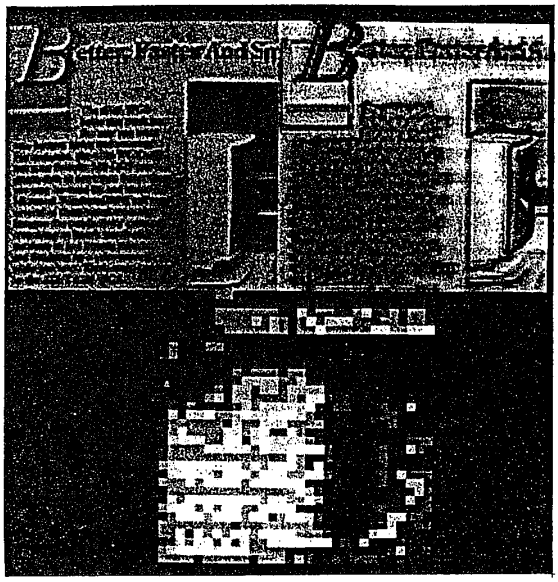


Figure 2: In the upper left is the zero sub-band. To its right is the perceptual threshold function The perceptual threshold function provides an measure of the sensitivity of each point to coding noise. Dark pixels indicate insensitive portion of the image, while white areas are very sensitive. The bottom row shows the activity measure. The number of sub-bands that are coded at each point is a function of the local frequency content. Black pixels denote that one sub-band was coded while white denotes that 6 sub-bands were coded.

5. *Noiseless Compression*

After quantization, the codewords for each sub-band are compressed using potentially multidimensional Huffman coding. First, if an entire subband consists of the zero codeword (which implies the perceptual threshold is met if every point in the sub-band is identical to zero), a tag notes this, and the coder proceeds to the next sub-band. If portions of a sub-band are non-zero, 4x4 blocks of zero codewords are identified. Depending on the percentage of *zero blocks*, one of two schemes of encoding this is used. If there are less than 102 non-zero blocks, the block number for each of these is sent, followed by the block's codewords. If there are more than 960 non-zero blocks, a bitmap is sent, followed by the codewords for the non-zero blocks. Smooth portions of an image require information from one sub-band. But, textured areas and edges, due to their broad spectrum, require information from several sub-bands. The perceptual threshold function automatically determines the number of sub-bands that must be coded. Shown below is a representation of this activity measure for the text image. Black pixels indicate that one sub-band was required at that point. Each successively lighter shade of gray denotes another sub-band was coded. For this image, a maximum of six sub-bands were required at any one point, even though portions of nine sub-bands were coded.

Sub-band #	% coded	% 4d	% 2d	% 1d
0	100.0	19.6	33.7	46.7
1	77.1	21.9	38.5	16.8
2	31.1	0.1	31.1	0.0
4	67.9	19.0	48.4	0.4
5	41.6	2.1	39.5	0.0
6	0.8	0.0	0.8	0.0
8	27.1	0.2	27.0	0.0
9	1.0	0.0	1.0	0.0
12	1.8	0.0	1.8	0.0

Table 3 - The coding algorithm encodes only the perceptually relevant portions of a sub-band. In addition, multidimensional Huffman coding is highly effective.

Each non-zero block is encoded using one, two, or four dimensional Huffman codebooks. The codebook with the highest dimensionality that will fit the rate (i.e. lowest potential rate) is used for each block. The dimensionality of the codebook for each block is combined with the block activity information and transmitted for each sub-band that is not all zeros. The four dimensional codebook operates on 2x2 codeword blocks, where each codeword has an absolute value of less than 4. The two dimensional codebook operates on 2x1 codeword blocks, where each codeword has an absolute value of less than 26. Likewise the one dimensional codebook operates on individual codewords, of any size required to meet the perceptual threshold. Since the quantizer outputs are entropy coded, and hence inherently of a variable bit length, the high peak quantizer outputs do not degrade the transmission cost of less active areas of the same image by a factor of $\log_2(\textit{largest level count})-\log_2(\textit{mean level count})$ as would happen in a standard DPCM coder.

6. *Testing and Results*

A wide selection of images, ranging from simple (low-resolution scenery) to complex (strongly contrasting textures, grey level text), have been collected for both training purposes and test purposes. No image is included in both the test and training sets. The results reported in this paper are for images that are in the test set, which consists of around 30 512x512 grey level images. All codebooks used in the compression algorithms were generated strictly from the training set, which consists of 107 images that are distinct from the test set.

The results of this compression algorithm provide an image quality, at a rate of .33 bit/pixel, for the "Lena Image" similar to or better than that of the .5 bit/pixel coder previously reported in ICASSP '88 [Safranek]. The quality of the Lena image is nearly transparent at 6x the image height at a rate of .5 bit/pixel. Using this algorithm, typical images require from .1 bit/pixel to .6 bit/pixel, and extremely complex textures require in the range of 0.9 bit/pixel for a high-quality encoding, or 1.5 bits/pixel for near-transparent coding. Grey scale text images that are not obviously impaired also require roughly .9 bits/pixel, while readable (for characters understandable in the original at 6x the image height) grey scale images

of text require about .5 bits/pixel Figures 4 to 7 present the output of this coder on a variety of images at three different quality levels. For these examples, the upper left image is the 8 bit/pixel original, the upper right image is at the nearly transparent quality level, the lower left offsets the perceptual threshold function by 5dB, and the lower right offsets the perceptual threshold function by 10 dB.

7. *Conclusions*

We have presented a variable bit rate coder which provides approximately constant quality for a wide range of input image complexities. Its compression gains are a result of a combination of all of the compression methods (DPCM, entropy coding, perceptual-threshold calculation, and quiescent block rejection), which work co-operatively to automatically provide good compression results and quality over a variety of images without user intervention.

References

— [Clarke] - *Transform Coding of Images* Academic Press, Inc. Orlando FL, 1985.

— [Cornsweet] - *Visual Perception* Academic Press, Inc. Orlando, FL, 1970.

— [Cox] - Cox, R. V., *The design of uniformly and nonuniformly spaced pseudo quadrature mirror filters*, IEEE Trans. ASSP, Vol ASSP-34, No. 5, pp. 1090-1096, October 1986.

— [Safranek] - Safranek, R. J., MacKay, K., Jayant, N. S., and Kim, T., *Image coding based on selective quantization of the reconstruction noise in the dominant sub-band.* Proc. IEEE ASSP88, New York, NY, Vol. M, pp. 765-768.

— [Woods] - Woods, J. W. and O'Neil, S. D., *Subband coding of images*, IEEE Trans. ASSP, Vol ASSP-34, No. 5, pp. 1278-1288, October 1986.

Figure 3: Here are the results of coding lena. The original image is in the upper left. The upper right is coded at 0.47 bits/pixel, lower left is coded at 0.33 bits/pixel, and the lower right is coded at 0.23 bits/pixel.

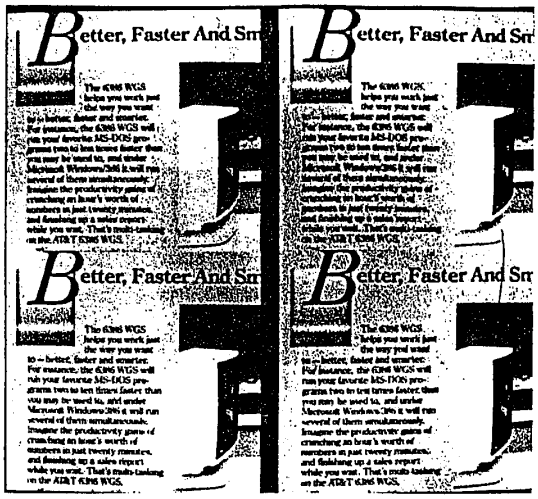


Figure 4: Here is an example of coding gray level text. The original image is in the upper left. The upper right is coded at 0.83 bits/pixel, lower left is coded at 0.47 bits/pixel, and the lower right is coded at 0.34 bits/pixel.

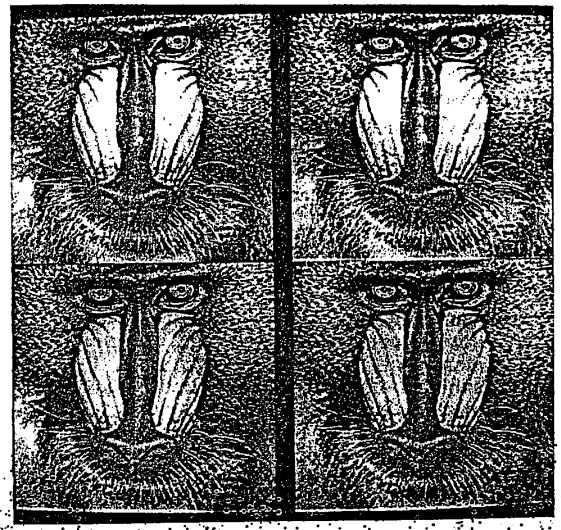


Figure 5: Here is an example of coding mandrill. The original image is in the upper left. The upper right is coded at 1.00 bits/pixel, lower left is coded at 0.58 bits/pixel, and the lower right is coded at 0.37 bits/pixel.

A JPEG Compliant Encoder Utilizing Perceptually Based Quantization

Robert J. Safranek
Signal Processing Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974
rjs@research.att.com

Abstract

Recently, the use of image compression algorithms in commercial products has been increasing an extremely fast rate. This explosion has been fueled by two recent developments, the availability of cheap signal processing ICs and the completion of several international standards for image compression. The ICs make the products cost effective, and the standards provide for a large degree of interoperability.

One of these standards, JPEG (Joint Photographics Experts Group), deals with the compression of still images. As is typical of these newly evolving standards, it specifies the information contained in the compressed bit stream, and a decoder architecture which can reconstruct an image from the data in the bit stream. However, the exact implementation of the encoder is not standardized. The only requirement on the encoder is that is generate a compliant bit stream. This provides an opportunity to introduce new research result

The challenge in improving these standards based codecs is to generate a compliant bitstream which produces a perceptually equivalent image as the baseline system that has a higher compression ratio. This results in a lower encoded bit rate without perceptual loss in quality. The proposed encoder uses the perceptual model developed by Johnston and Safranek [JohnSaf] to determine, based on the input data, which coefficients are perceptually irrelevant. This information, is used to remove (zero out) some coefficients before they are input to the quantizer block. This results in a larger percentage of zero codewords at the output of the quantizer which reduces the entropy of the resulting codewords.

1. Introduction

Recently, the use of image compression algorithms in commercial products has been increasing rapidly. This explosion has been fueled by two recent developments, the availability of cheap signal processing ICs and the establishment of several international standards for image compression. The ICs make the products cost effective, and the standards provide for a large degree of interoperability.

One of these standards, JPEG (Joint Photographics Experts Group), deals with the compression of still images. Some applications in which it has been utilized are archival storage of images for the publishing industry, reducing storage requirements for picture archiving systems, and ISDN based image services. In addition, it has been used for intraframe only compression of motion video.

As is typical of these newly evolving standards, it specifies the information contained in the compressed bit stream, and a decoder architecture which can reconstruct an image from the data in the bit stream. However, the exact implementation of the encoder is not standardized. The only requirement on the encoder is that it generates a compliant bit stream. This provides an opportunity for people to improve the compression efficiency and/or subjective image quality by designing better encoders. This paper will present an such encoder for the Baseline Sequential

0-8194-1474-3/94/$6.00

As shown in Figure 1, the encoder consists of three major components, a Forward Transform, Quantization, and Entropy Coding. The Forward Transform is an 8x8 Discrete Cosine Transform (DCT). Its purpose is to reduce number of samples that need to be transmitted by performing energy compaction on the signal. Since most images have a low pass spectrum, transforming the spatial domain data into the frequency domain results in a fewer significant samples. In addition these samples tend to be clustered at the low frequencies.

The purpose of the quantization step is to take the raw output of the DCT and quantize the coefficients. This step results in a loss of information, but provides for the majority of the data rate reduction in the system. By adjusting parameters in this stage, it is possible to control the compressed bitrate and output image quality.

Entropy Coding takes the fixed length quantized DCT coefficients and produces a set of variable length channel symbols. This operation attempts to produce a compressed data stream whose rate is as close as possible to the entropy of the quantized DCT coefficients.

2.1 Quantization

We will now focus on how the quantization is performed since that step is vital in understanding the improved encoder. The forward DCT produces 64 coefficients. These coefficients are then uniformly quantized. The quantizer step size that is used for each coefficient is determined by a Quantization Table which must be specified by the application as an input to the encoder. Elements in the Quantization Table can take on integer values in the range of 1 to 255.

The quantization process is defined as a division of each DCT coefficient by its corresponding entry from the quantization table, followed by rounding to the nearest integer.

$$F_Q(u,v) = IntegerRound\left[\frac{F(u,v)}{Q(u,v)}\right]$$

where $F(u,v)$ the DCT coefficients for a given input block, $F_Q(u,v)$ are the quantized DCT coefficients, and $Q(u,v)$ is the Quantization Table.

In the decoder, the inverse operation is performed which provides the decoder with the values appropriate for input to the inverse DCT.

$$F_{Q'}(u,v) = F_Q(u,v) * Q(u,v)$$

where F sub Q' (u,v) are the reconstructed DCT coefficients for a given block.

From this discussion it is clear that the Quantization Table is part of the information that must be transmitted from the encoder to decoder. If an entry in the Quantization Table is greater than unity, information loss occurs. The table is chosen to trade off compression efficiency and subjective image quality.

3. Perceptual Model

In has long been known that the human visual system is not an ideal receiver, and that it is possible to take advantage of this fact in the encoding process [Cornsweet]. It has only been recently however, that more systematic investigation of the use of visual masking in image compression has occurred [JaJoSa]. These studies have attempted to derive a computational

proposed by Watson do not have this restriction and provide more adaptability at the cost of increased computational load.

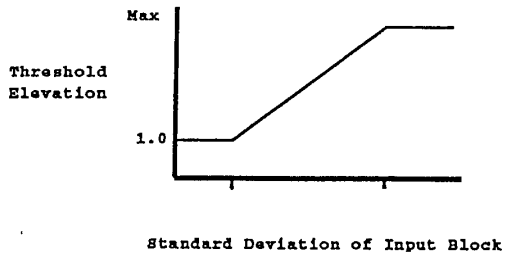


Figure 2: Presented here is an example of a threshold elevation curve.

4. Perceptual Encoder

The previous section described a method for determining a set of masking thresholds for each block of an input image which result in a unique perceptually optimal quantization table for each block. Unfortunately, JPEG allows only one quantization table for each image. Therefore the problem that must be solved is how to make use of this local information within the framework of the JPEG standard. If you examine the forward quantization equation in section 2.1, it is clear that all input coefficients that have a value less than their corresponding quantization table entry will be quantized to a value of zero. This observation is the key to incorporating locally adaptive quantization into JPEG.

Since a quantized coefficient with a value of zero is a valid member of the JPEG bitstream, the perceptually based encoder will identify which coefficients can be set to zero while maintaining the subjective quality of the encoded image. This will maintain complaince with the JPEG bitstream specification while reducing the bitrate required to encode the image.

Figure 3 illustrates the structure of such an encoder. The forward transform is identical to the one in baseline JPEG. At this point, the DCT coefficients are input to the perceptual model which generates the data dependent quantization table for that block. This table and the raw DCT coefficients are now input to a "pre-quantizer." The purpose of this module is to zero out the coefficients that have a magnitude less than the corresponding entry in the quantization table for that block, and pass the other coefficients through unchanged.

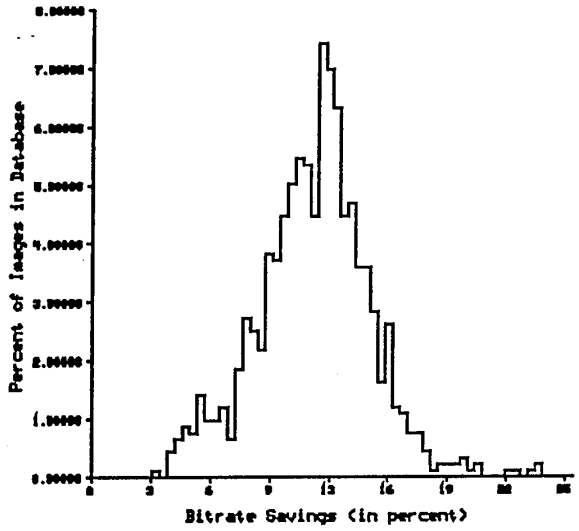


Figure 4: Bitrate savings over baseline JPEG obtained from PxJPEG

Two objective evaluation methods were used. The first was a traditional A/B forced choice test. The subject was simultaneously shown two images sided by side on a video monitor. One was the original image and the other was the same image encoded using either JPEG or PxJPEG. The order of presentation, that is which side the original image was located on, was randomized. Given this stimulus, the task was to determine which image was the original. A test set of 10 images that was used. This set was chosen to contain typical images, as well as test patterns that would stress the encoder. At present, this test has been taken by 7 times by a single subject, the author, who was familiar with the test data. The result of this test was that both JPEG and PxJPEG using the perceptually optimal quantization matrices were statistically indistinguishable from the original image.

In order to provide further insight into the subjective quality of the codecs, the output images were evaluated using Scot Daly's Visual Difference Predictor (VDP) [Daly]. This algorithm takes as input two images, a reference and a test, as well as viewing condition and a characterization of the display. The input images are normalized to account for the viewing conditions and display, and then passed thought a detailed model of the human visual system. It

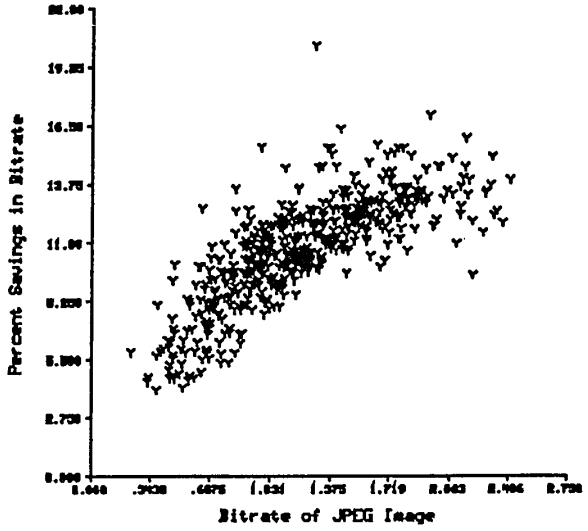


Figure 5: Bitrate savings by PxJPEG as a function of JPEG bitrate

7. Acknowledgements

I would like to thank Jim Johnston for our many discussions on perceptual coding and masking, the members of the image group for their many inputs, Scot Daly for answering my questions on the VPD, the Independent JPEG group (IJG) and Tom Lane for providing a JPEG implementation, Edmund Yeh for implementing most of the VPD algorithm, and James Pawlyk for keeping the lab running.

References

— [Cornsweet] - *Visual Perception* Academic Press, Inc. Orlando, FL, 1970.

— [Daly] - Daly, S. "The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity" in *Digital Images and Human Vision,* edited by Andrew B. Watson, MIT Press, Cambridge, MA, London, England 1993.

DCT quantization matrices visually optimized for individual images

Andrew B. Watson

MS 262-2 NASA Ames Research Center
Moffett Field, CA 94035-1000
beau@vision.arc.nasa.gov

ABSTRACT

Several image compression standards (JPEG, MPEG, H.261) are based on the Discrete Cosine Transform (DCT). These standards do not specify the actual DCT quantization matrix. Ahumada & Peterson[1] and Peterson, Ahumada & Watson[2] provide mathematical formulae to compute a perceptually lossless quantization matrix. Here I show how to compute a matrix that is optimized for a particular image. The method treats each DCT coefficient as an approximation to the local response of a visual "channel." For a given quantization matrix, the DCT quantization errors are adjusted by contrast sensitivity, light adaptation, and contrast masking, and are pooled non-linearly over the blocks of the image. This yields an 8x8 "perceptual error matrix." A second non-linear pooling over the perceptual error matrix yields total perceptual error. With this model we may estimate the quantization matrix for a particular image that yields minimum bit rate for a given total perceptual error, or minimum perceptual error for a given bit rate. Custom matrices for a number of images show clear improvement over image-independent matrices. Custom matrices are compatible with the JPEG standard, which requires transmission of the quantization matrix.

1. JPEG DCT QUANTIZATION

The JPEG image compression standard provides a mechanism by which images may be compressed and shared among users [3, 4]. I briefly review the quantization process within this standard. The image is first divided into blocks of size {8,8}. Each block is transformed into its DCT, which we write c_{ijk}, where i,j indexes the DCT frequency (or basis function), and k indexes a block of the image. Though the blocks themselves form a two dimensional array, for present purposes a one dimensional block index is sufficient. Each block is then quantized by dividing it, coefficient by coefficient, by a quantization matrix (QM) q_{ij}, and rounding to the nearest integer

$$u_{ijk} = \text{Round}\left[c_{ijk}/q_{ij}\right] \quad . \tag{1}$$

The quantization error e_{ijk} in the DCT domain is then

$$e_{ijk} = c_{ijk} - u_{ijk}\, q_{ij} \quad . \tag{2}$$

2. IMAGE-INDEPENDENT PERCEPTUAL QUANTIZATION

The JPEG QM is not defined by the standard, but is supplied by the user and stored or transmitted with the compressed image. The principle that should guide the design of a JPEG QM is that it provide optimum visual quality for a given bit rate. QM design thus depends upon the visibility of quantization errors at the various DCT frequencies. In recent papers, Peterson *et al.* [5, 6] have provided measurements of threshold amplitudes for DCT basis functions. For each frequency ij they measured psychophysically the smallest coefficient that yielded a visible signal. Call this threshold t_{ij}. From Eqn.s (1) and (2) it is clear that the maximum possible quantization error e_{ijk} is $q_{ij}/2$. Thus to ensure that all errors are invisible (below threshold), we set

$$q_{ij} = 2\, t_{ij} \quad . \tag{3}$$

0-8194-1146-9/93/$6.00

I call this the Image-Independent Perceptual approach (IIP). It is perceptual because it depends explicitly upon detection thresholds for DCT basis functions, but is image-independent because a single matrix is computed independent of any image. Ahumada *et al.* [1,7] have extended the value of this approach by measuring t_{ij} under various conditions and by providing a formula that allows extrapolation to other display luminances (L) and pixel sizes (px,py), as well as other display properties. For future reference, we write this formula in symbolic form as

$$t_{ij} = \mathrm{ap}[i, j, L, px, py, \ldots] \tag{4}$$

3. LIMITATIONS OF THE IIP APPROACH

While a great advance over the *ad hoc* matrices that preceded it, the IIP approach has several shortcomings. The fundamental drawback is that the matrix is computed independent of the image. This would not be a problem if visual thresholds for artifacts were fixed and independent of the image upon which they are superimposed, but this is not the case.

First, visual thresholds increase with background luminance. The formula of Ahumada & Peterson describes the threshold for DCT basis functions as a function of a mean luminance. This would normally be taken as the mean luminance of the display. But variations in local mean luminance within the image will in fact produce substantial variations in DCT threshold. We call this *luminance masking*.

Second, threshold for a visual pattern is typically reduced in the presence of other patterns, particularly those of similar spatial frequency and orientation, a phenomenon usually called *contrast masking*. This means that threshold error in a particular DCT coefficient in a particular block of the image will be a function of the value of that coefficient in the original image.

Third, the IIP approach ensures that any single error is below threshold. But in a typical image there are many errors, of varying magnitudes. The visibility of this error ensemble is not generally equal to the visibility of the largest error, but reflects a pooling of errors, over both frequencies and blocks of the image. I call this *error pooling*.

Fourth, when all errors are kept below a perceptual threshold a certain bit rate will result. The IIP method gives no guidance on what to do when a lower bit rate is desired. The ad hoc "quality factors" employed in some JPEG implementations, which usually do no more than multiply the quantization matrix by a scalar, will allow an arbitrary bit rate, but do not guarantee (or even suggest) optimum quality at that bit rate. I call this the problem of *selectable quality*.

Here I present a general method of designing a custom quantization matrix tailored to a particular image. This *image-dependent perceptual* (IDP) method incorporates solutions to each of the problems described above: luminance masking, contrast masking, error pooling, and selectable quality. The strategy is to develop a very simple model of perceptual error, based upon DCT coefficients, and to iteratively estimate the quantization matrix which yields a designated perceptual error.

4. LUMINANCE MASKING

Detection threshold for a luminance pattern typically depends upon the mean luminance of the local image region: the brighter the background, the higher the luminance threshold [8,9]. This is usually called "light adaptation," but here we call it "luminance masking" to emphasize the similarity to contrast masking, discussed in the next section.

To illustrate this effect, the solid lines in Fig. 1 plot values of the formula for t_{ij} provided by Ahumada and Peterson[1] as a function of the mean luminance of the block, assuming that the maximum display luminance is 100 cd m^{-2} and that the greyscale resolution is 8 bits. The three curves are for five representative frequencies. These

curves illustrate that variations by as much as 0.5 log unit in t_{ij} might be expected to occur within an image, a variations in the mean luminance of the block.

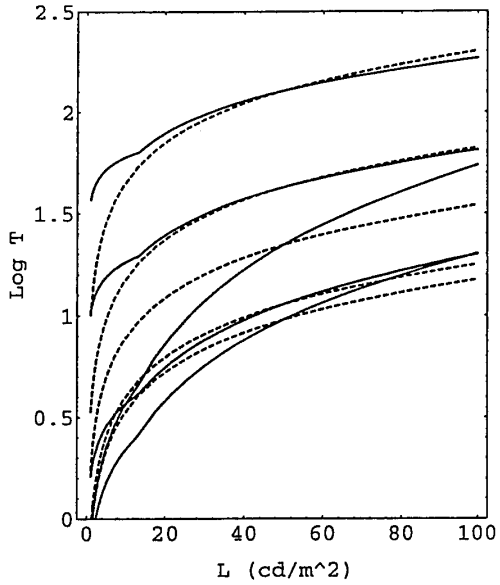


Figure 1. Log of t_{ij} as a function of luminance L of the block. From the top, the curves are for frequencies of {7 {0,7}, {0,0}, {0,3}, and {0,1}. The maximum display luminance is assumed to be 100 cd m^{-2}. The dashed curves are the power function approximation described in the text.

The effect of mean luminance upon the DCT thresholds is complex, involving both vertical and horizonta shifts of the contrast sensitivity function. We can compute a luminance-masked threshold matrix for each bloc either of two ways. The first is to make use of a formula such as that supplied by Ahumada and Peterson [1],

$$t_{ijk} = \mathrm{ap}[i, j, L_0\, c_{00k}/\bar{c}_{00}]$$

where c_{00k} is the DC coefficient of the DCT for block k., L_0 is the mean luminance of the display, and $\bar{c}_{00}$ is the coefficient corresponding to L_0 (1024 for an 8 bit image). This solution is as complete and accurate as the underlying formula, but may be rather expensive to compute. For example, in the *Mathematica* language, usin compiled function, and running on a SUN Sparc 2, it takes about 1 second per block.

A second, simpler solution is to approximate the dependence of t_{ij} upon c_{00k} with a power function:

$$t_{ijk} = t_{ij}\left(c_{00k}/\bar{c}_{00}\right)^{a_T} \quad .$$

The initial calculation of t_{ij} should be made assuming a display luminance of L_0 The parameter a_T takes name from the corresponding parameter in the formula of Ahumada and Peterson, wherein they suggest a va of 0.649. Note that luminance masking may be suppressed by setting a_T=0. More generally, a_T controls the degree to which this masking occurs. Note also that the power function makes it easy to incorporate a non-un display Gamma, by multiplying a_T by the Gamma exponent (see Section 10.2).

As illustrated by the dashed lines in Fig. 1, this power function approximation is accurate over an upper range of luminances (for the parameters in Fig. 1, above about 10 cd m^{-2}). Except for very dark sections of an image, this range should be adequate. The discrepancy is also greatest at the lowest frequencies, especially the DC term. This could be corrected by adopting a matrix of exponents, one for each frequency. But note that the discrepancy is a conservative one, that is the threshold changes less with block luminance than the model calls for. This may not be a bad thing, especially at DC, where the validity of the model may be least.

5. CONTRAST MASKING

Contrast masking refers to the reduction in the visibility of one image component by the presence of another. This masking is strongest when both components are of the same spatial frequency, orientation, and location. Here we consider only masking within a block and a particular DCT coefficient (It is possible to extend these ideas to masking between DCT coefficients, and across DCT blocks). We employ a model of visual masking that has been widely used in vision models, based on seminal work by Legge and Foley [10, 11]. Given a DCT coefficient c_{ijk} and a corresponding absolute threshold t_{ijk} our masking rule states that the masked threshold m_{ijk} will be

$$m_{ijk} = \mathrm{Max}\left[t_{ijk} , \left| c_{ijk} \right|^{w_{ij}} t_{ijk}^{\,1-w_{ij}} \right] \tag{7}$$

where w_{ij} is an exponent that lies between 0 and 1. Because the exponent may differ for each frequency, we allow a matrix of exponents equal in size to the DCT. Note that when w_{ij} =0, no masking occurs, and the threshold is constant at t_{ijk}. When w_{ij} = 1, we have what is usually called "Weber Law" behavior, and threshold is constant in log or percentage terms (for $c_{ijk} > t_{ijk}$). The function is pictured for a typical empirical value of w_{ij} = 0.7 in Fig. 2.

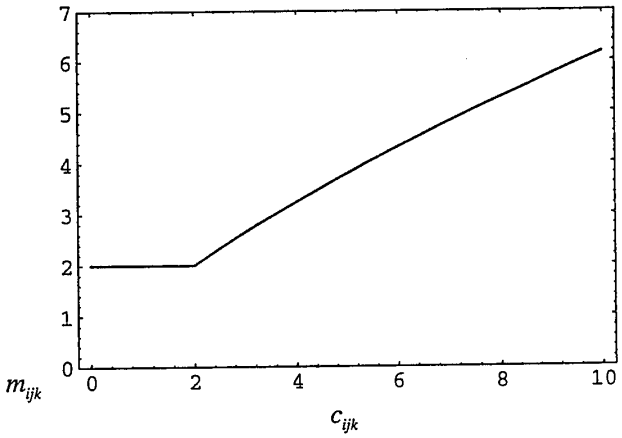


Figure 2. Contrast masking function, describing the masked threshold m_{ijk} as a function of DCT coefficient c_{ijk} , for parameters w_{ij}=0.7, t_{ijk} = 2.

Because the effect of the DC coefficient upon thresholds has already been expressed by luminance masking, we specifically exclude the DC term from the contrast masking, by setting the value of $w_{00} = 0$. It is interesting that while contrast masking is assumed to be independent from coefficient to coefficient (frequency to frequency), in the case of luminance masking the DC frequency affects *all* other frequencies.

Figure 3 shows the masked sensitivity (m_{ijk}^{-1}) for the Lena image. Note that the dark strip in the upper right results in generally higher sensitivity due to luminance masking (un-masking, perhaps we should say).

Figure 3. The Lena image and its masked sensitivity DCT (m_{ijk}^{-1}) for w_{ij}=0.7 and a_T=0.649. If w_{ij}=0 and a_T=0, cells would be identical and would look like the inset (2 t_{ij}).

6. PERCEPTUAL ERROR AND JUST-NOTICEABLE-DIFFERENCES

In vision science, we often express the magnitude of a signal in multiples of the threshold for that signal. These threshold units are often called "just-noticeable differences," or *jnd*'s. Having computed a masked thresho. m_{ijk}, the error DCT may therefore be expressed in jnd's as

$$d_{ijk} = e_{ijk} / m_{ijk} \tag{8}$$

Each value of d_{ijk} is an error in a particular frequency and block, expressed as a proportion of the just-detectable error in that frequency and block. Thus all the errors are now in the "common coin" of perceptual erro: the *jnd*.

7. SPATIAL ERROR POOLING

To pool the errors in the jnd DCT we employ another standard feature of current vision models: the so-callec β-norm (or Minkowski metric). It often arises from an attempt to combine the separate probabilities that individual errors will be seen, in the scheme known as "probability summation" [12, 13, 14]. We pool the jnds for a particular frequency $\{i,j\}$ over all blocks k as

$$p_{ij} = \left(\sum_k \left| d_{ijk} \right|^{\beta_s} \right)^{1/\beta_s} \tag{9}$$

Different values of the exponent β_s implement different types or degrees of pooling. When β_s=1, the pooling is linear summation of absolute values. When β_s=2, the errors combine quadratically, in an RMS or standard deviation type measure. When $\beta_s = \infty$(in practice, a large number such as 100 will do), the pooling rule becomes a maximum-of operation: only the largest error matters. In psychophysical experiments that examine

summation among sinusoidal components of differing frequency, a β_s of about 4 has been observed [15, 16, 17]. The exponent β_s is given here as a scalar, but may be made a matrix equal in size to the QM to allow differing pooling behavior for different DCT frequencies. This matrix p_{ij} of "pooled jnds" is now a simple measure of the visibility of artifacts within each of the frequency bands defined by the DCT basis functions. I call it the "perceptual error matrix."

8. FREQUENCY ERROR POOLING

This perceptual error matrix p_{ij} may itself be of value in revealing the frequencies that result in the greatest pooled error for a particular image and quantization matrix. But to optimize the matrix we would like a single-valued perceptual error metric. We obtain this by combining the elements in the perceptual error matrix, using a Minkowski metric with a possibly different exponent, β_f

$$P = \left(\sum_{ij} p_{ij}^{\beta_f} \right)^{1/\beta_f} . \tag{10}$$

It is now straightforward, at least conceptually, to optimize the quantization matrix to obtain minimum bit-rate for a given P, or minimum P for a given bit rate. In practice, however, a solution may be difficult to compute. But if $\beta_f = \infty$, then P is given by the maximum of the p_{ij}. Under this condition minimum bit-rate for a given $P = \psi$ is achieved when all $p_{ij} = \psi$. Intuitively, if the maximum of the p_{ij} equals ψ, each of the others might as well be increased to ψ, since that will not increase P, but will decrease bit-rate.

Recall that each entry in the matrix p_{ij} corresponds (at least monotonically) with the visibility of a particular class of artifact: that of the corresponding frequency (basis function). This strategy of equating all p_{ij} to ψ thus also has the effect of equating the visibilities of each of these classes of error.

While it is likely that the true value of β_f is nearer to β_s (approximately 4), it also seems likely that this more accurate value will not greatly alter the outcome of the optimization and will not be worth the substantial increase in computational effort.

8. OPTIMIZATION METHOD

Under the assumption $\beta_f = \infty$, the joint optimization of the quantization matrix reduces to the vastly simpler separate optimization of the individual elements of the matrix. Each entry of the perceptual error matrix p_{ij} may be considered an independent function of the corresponding entry q_{ij} of the quantization matrix

$$p_{ij} = \mathrm{f}_{ij}\left(q_{ij}\right) . \tag{11}$$

This function is monotonically increasing and

$$f_{ij}(1) = 0 \quad \forall \; i, j . \tag{12}$$

We seek a particular $\hat{q}_{ij}$ such that

$$f_{ij}\left(\hat{q}_{ij}\right) = \psi \quad \forall \; i, j . \tag{13}$$

Of course, in some cases no amount of quantization will yield a value as large as the target ψ (for example, if all coefficients are quantized to 0, but the error remains below ψ). For those cases we are content to set $\hat{q}_{ij}$ to an arbitrary maximum, such as 255 (the largest quantization table entry permitted in the JPEG baseline standard).

In a practical implementation, a rapid method of estimating $\hat{q}_{ij}$ is required. Here we have used a bisection method that, while slow, is guaranteed to find a solution. A range is established for q_{ij} between lower and upp bounds of $\overset{<}{q}_{ij}$ and $\overset{>}{q}_{ij}$ (typically {1,255}). p_{ij} is evaluated at the midpoint of the range,

$$\overline{q}_{ij} = \text{Round}\left[\frac{1}{2}\left(\overset{<}{q}_{ij} + \overset{>}{q}_{ij}\right)\right] . \qquad (1$$

If $p_{ij} < \psi$, then $\overset{<}{q}_{ij} = \overline{q}_{ij}$, otherwise, $\overset{>}{q}_{ij} = \overline{q}_{ij}$. This procedure is repeated until $\overline{q}_{ij}$ no longer changes. As practical matter, since QM's in baseline JPEG are eight bit integers, this degree of accuracy is obtained in n=9 iterations from a starting range of 255 .

In the following examples, unless otherwise stated, the parameter values used were $a_T = 0.649$, $\beta = 4$, w_{ij} 0.7 , display mean luminance L_0= 65 cd m^{-2} , image greylevels = 256, $\overline{c}_{00}$ = 1024. The viewing distance was assumed to yeild 32 pixels/degree. For a 256 by 256 pixel image, this corresponds to a viewing distance if 7.115 picture heights. The "JPEG bit rate" is calculated by computing the code size for AC and DC coefficients using t default JPEG Huffman tables. It does not include the overhead composed of quantization tables, Huffman tabl marker codes, etc. because this overhead is not image dependent and depends on coding decisions made by th application (e.g. use of restart intervals). If it had been included it would increase the bit rate for a 256 by 256 image by about 0.038 bits/pixel.

Several steps in the iterative estimation of $\hat{q}_{ij}$ are illustrated in Fig. 4. Successive steps show further refinement in $\hat{q}_{ij}$, and a progressively more uniform matrix p_{ij}. On step 1, $q_{ij} = 255$, $\forall$ i, j. On this step th perceptual error matrix shows greatest error at low spatial frequencies.

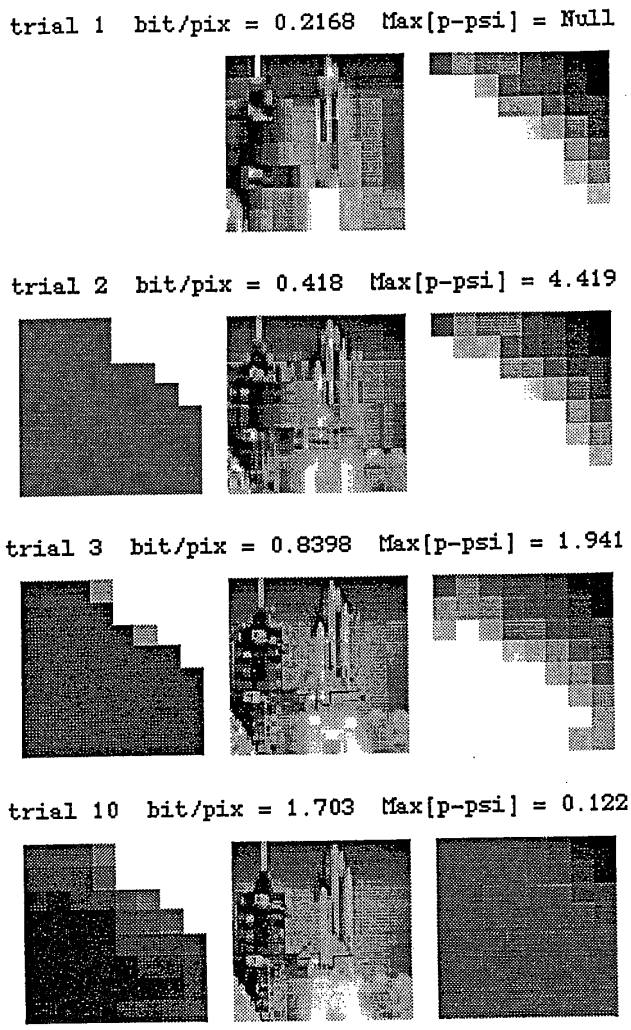


Figure 4. Iterative estimation of the custom quantization matrix $\hat{q}_{ij}$. The three panels in each row show quantization matrix q_{ij}, the reconstructed image using q_{ij}, and the perceptual error matrix p_{ij}. The labels indicate the iteration trial, the current JPEG bit-rate, and the maximum difference between p_{ij} and ψ (discounting those for which the maximum error is always less than ψ). The image was {64,64}, target ψ was 1. For q_{ij} and p_{ij}, the DC coefficient is at the lower left corner.

Figure 5 shows the Lena image [18] compressed to various values of perceptual error ψ = {1, 2, 4, 8}. The value of ψ=1 produces an essentially "perceptually lossless" compression [19] under the prescribed viewing conditions (mean luminance = 65 cd m^{-2}, 32 pixels/deg.

Figure 5. The Lena image compressed using custom matrices designed for perceptual error levels (ψ) of 1, 2, 4, and 8. Corresponding bit rates were 2.28, 1.47, 0.72, 0.24. The original image had dimensions of {256,256}.

It is interesting to compare the image-independent quantization matrix to the custom matrix for various quality levels. This is shown in Table 1, where we give the ratio of image-dependent and independent matrices, for two quality levels of 1 and 4. Elements that have been set to the maximum of 255 are indicated by zeros. Note that image dependence does alter the structure of the matrix, and that changes in quality (as defined here) do not yield a constant scaling of the basic matrix.

0.156	0.231	0.193	0.208	0.192	0.165	0.172	0.161
0.231	0.223	0.208	0.179	0.182	0.167	0.146	0.155
0.193	0.208	0.166	0.174	0.171	0.157	0.156	0.171
0.162	0.179	0.174	0.165	0.154	0.158	0.166	0.194
0.157	0.141	0.171	0.154	0.156	0.166	0.164	0.243
0.165	0.167	0.157	0.158	0.195	0.208	0.226	0.289
0.172	0.168	0.156	0.181	0.198	0.235	0.317	
0.187	0.171	0.158	0.217	0.251			

0.615	1.41	1.24	1.13	1.07	1.46	
1.1	1.11	1.28	1.04	1.15	1.38	3.28
1.07	1.1	1.11	1.22	1.28	1.55	
1.04	1.1	1.26	1.25	1.72		
1.55	1.39	1.69	2.05			

Table 1. Ratio of image-dependent and independent quantization matrices for the Lena image at quality levels of 1 (top) and 4 (bottom). This ratio is equal to $\hat{q}_{ij}/2t_{ij}$. Empty cells indicate that the image-dependent matrix had a value of 255 (the maximum allowed).

9. OPTIMIZING QM FOR A GIVEN BIT-RATE

It is of interest to relate the JPEG bit-rate to the perceptual error level ψ. This is shown for the Lena and Mandrill images in Fig. 6. This is a sort of inverse "rate-distortion" function. Note that useful bit-rates below 2 bits/pixel yield perceptual errors above about 2.

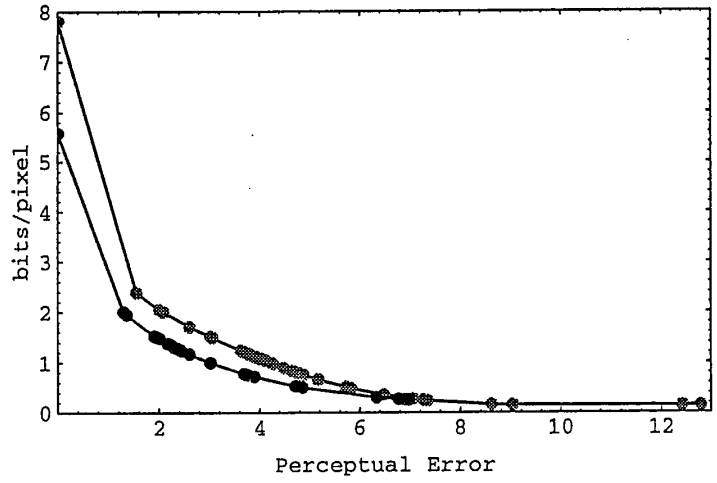


Figure 6. JPEG bit-rate versus perceptual error ψ for the Lena (lower curve) and Mandrill (upper curve) images. The lines are second order polynomial interpolations.

The method described above yields a QM with a specified perceptual error ψ. However, one may desire a QM that yields a given bit rate h_0 with minimum perceptual error ψ. This can be done iteratively by noting that the bit rate is a decreasing function of ψ, as shown in Fig. 6. In our current implementation, we use a second order interpolating polynomial fit to all previous estimated values of $\{h, \psi\}$ to estimate the next candidate ψ, terminating when $|h - h_0| < \Delta h$, where Δh is the desired accuracy in bit-rate. On each iteration, a complete estimation of $\hat{q}_{ij}$ is performed. There are no doubt more rapid methods.

The most meaningful contest between IDP and IIP approaches is to compare images compressed by the two methods to a constant bit rate. Furthermore, the bit rate must be low enough that the poorer method shows visible artifacts, else both will appear perfect. Figures 7 and 8 provide such comparisons. The IDP method is visibly superior, even in relatively low-quality printed renditions.

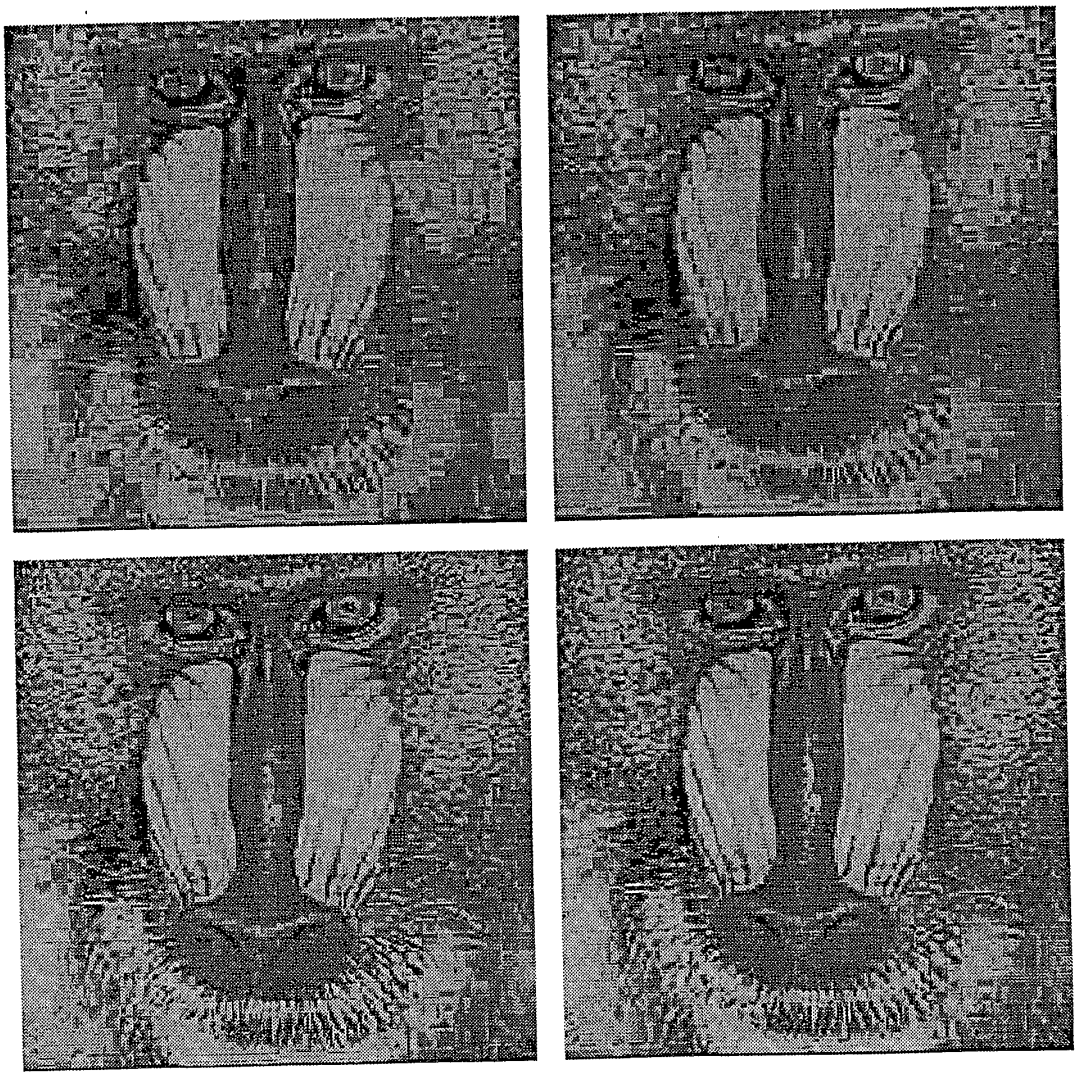


Figure 7. IIP (left) and IDP (right) compressions at 0.25 bits/pixel (top row) and 0.5 bits/pixel (bottom row).

Figure 8. IIP (left) and IDP (right) compressions at 0.25 bits/pixel (top row) and 0.5 bits/pixel (bottom row).

10. EXTENSIONS AND FUTURE RESEARCH

10.1 Estimation of t_{ij} w_{ij}, β_s, a_T

The method described here depends upon estimates of the matrices t_{ij} and w_{ij}, and the parameters β_s and a_T. Estimates of t_{ij} may be obtained directly from psychophysical experiments that measure detection thresholds for individual DCT basis functions [1, 5, 6]. We are devising experiments, adapted from the methods of Legge and Foley [10, 11] to directly estimate w_{ij}. In these experiments detection thresholds are measured for an increment (or

decrement) in the amplitude of a DCT basis function. Estimation of β_s is more difficult. Several values of β_s in the range of 1-100 could be evaluated for the degree to which they yield a plausible perceptual error metric p_{ij}. In addition, a matrix of values of β_s might be warranted, with different degrees of spatial pooling at each DCT frequency.

10.2 Gamma Functions

Remarkably, the JPEG specification makes no statement regarding the relation between pixel values and displayed luminance. While one can understand their reluctance to impose constraints upon JPEG applications, it should be understood that ultimate visual quality depend on this relation. The "de facto" assumption appears to be that pixel values will be applied directly to the display subsystem, which typically has a non-linear relation between greylevel and luminance, often known as a "gamma function" that is approximately a power function with an exponent (gamma) of about 2.3. The assumption presumably also is that variations in this function from system to system are not so great as to seriously degrade visual quality.

In an ideal system, one would specify both the gamma function of image capture, and of the target display. Image data would be transformed to luminance before compression, and after reconstruction, to values that would result in luminance on the display. Unfortunately, we cannot add descriptors of these gamma functions to the existing JPEG specification, so we must be content with the "de facto" assumption.

Since the preceding calculations have treated pixel values as proportional to luminance (gamma=1), under the "de facto" assumption, we should subject the image data to inverse and forward gamma transformations before coding and after decoding, respectively. The present approach, which does no such transformations, relies on the approximate linearity of the gamma function near the middle of its range, and on the inclusion of the display gamma into the luminance masking function as discussed in Section 4. This subject will be examined in future research.

10.3 Color Images

The Image-Dependent Perceptual approach has been described here only with respect to coding of monochrome images. The principles, however, are easily extended to color images. The simplest approach is to measure or compute a unique t_{ij} for each of the three color channels[7], and from them compute three custom quantization matrices. The matter may be complicated by different masking and pooling properties in the chromatic channels than in the luminance channel. But since color consumes so small a part of the total bit-rate, these details are not likely to be critical in practical applications.

11. SUMMARY

I have shown how to compute a visually optimal quantization matrix for a given image. These image-dependent quantization matrices produce better results than image independent matrices. The algorithm can be easily incorporated into JPEG compliant applications.

In a practical sense, the IDP method proposed here solves two problems. The first is to provide maximum visual quality for a given bit rate. The second problem it solves is to provide the user with a sensible and meaningful quality scale for JPEG compression. Without such a scale, each image must be repeatedly compressed, reconstructed, and evaluated by eye to find the desired level of visual quality.

However, at present, it is admittedly only a conjecture that this scale relates in a direct way to perceived visual quality. While I am confident that it relates more directly to quality than does the ad hoc "quality factor" of some JPEG implementations, to demonstrate a robust relation between computed perceptual error and perceived quality will require subjective judgments, both over different bit rates and different images.

From the standpoint of computational complexity, this algorithm adds only a modest amount to the cost of JPEG image compression. All optimization takes place in the DCT domain, so no additional forward or inverse DCT's are required. The DCT mask is computed only once, and consists of a few calculations on each DCT pixel. The estimation of the quantization matrix requires a maximum of ten (and probably many fewer) iterations, each of which consists of a modest number of simple operations on each DCT pixel. It is certainly a smaller burden than requiring the user to repeatedly compress, reconstruct, and visually assess the result.

12. NOTATION

c_{ijk}	DCT of an image
q_{ij}	quantization matrix
u_{ijk}	quantized DCT
e_{ijk}	DCT error
t_{ij}	DCT threshold matrix (based on global mean luminance)
$\mathrm{ap}[i,j,L,px,py,\ldots]$	threshold formula of Ahumada and Peterson[1]
t_{ijk}	DCT threshold matrix (based on local mean luminance c_{00k})
a_T	luminance masking exponent
w_{ij}	contrast masking exponent (Weber exponent)
m_{ijk}	mask DCT
d_{ijk}	jnd DCT
p_{ij}	perceptual error matrix
β_s	spatial error-pooling exponent
P	total perceptual error
β_f	frequency error-pooling exponent
c_{00k}	DC coefficient in block k
L_0	mean luminance of the display
$\bar{c}_{00}$	Average DC coefficient, corresponding to L_0 (typically 1024)
ψ	target total perceptual error value
$\hat{q}_{ij}$	estimated quantization matrix yielding target perceptual error

13. ACKNOWLEDGMENTS

I thank Albert J. Ahumada, Jr. and Heidi A. Peterson for valuable discussions. This work was supported by NASA RTOPs 506-59-65 and 505-64-53.

14. REFERENCES

1. A. J. Ahumada Jr. and H. A. Peterson. "Luminance-Model-Based DCT Quantization for Color Image Compression," in *Human Vision, Visual Processing, and Digital Display III*, B. E. Rogowitz, ed. (Proceedings of the SPIE, 1992).

2. H. A. Peterson, A. J. Ahumada Jr. and A. B. Watson. "The Visibility of DCT Quantization Noise," SID Digest of Technical Papers, in press (1993).

3. G. Wallace. "The JPEG still picture compression standard," Communications of the ACM. 34(4), 30-44 (1991).

4. W. B. Pennebaker and J. L. Mitchell. *JPEG Still image data compression standard* (Van Nostrand Reinhold, New York, 1993).

5. H. A. Peterson. "DCT basis function visibility in RGB space," in *Society for Information Display Digest of Technical Papers*, J. Morreale, ed. (Society for Information Display, Playa del Rey, CA, 1992).

6. H. A. Peterson, H. Peng, J. H. Morgan and W. B. Pennebaker. "Quantization of color image components in the DCT domain,". Human Vision, Visual Processing, and Digital Display. Proc. SPIE. **1453**: 210-222, 1991.

7. H. A. Peterson, A. J. Ahumada Jr. and A. B. Watson. "An improved detection model for DCT coefficient quantization," SPIE Proceedings. **1913** (In press), (1993).

8. F. L. van Nes and M. A. Bouman. "Spatial modulation transfer in the human eye," Journal of the Optical Society of America. **57**, 401-406 (1967).

9. H. B. Barlow. "Dark and light adaptation: Psychophysics," in *Handbook of Sensory Physiology*, L. Hurvich and D. Jameson, ed. (Springer-Verlag, New York, 1972).

10. G. E. Legge and J. M. Foley. "Contrast masking in human vision," Journal of the Optical Society of America. 70(12), 1458-1471 (1980).

11. G. E. Legge. "A power law for contrast discrimination," Vision Research. **21**, 457-467 (1981).

12. N. Graham. "Visual detection of aperiodic spatial stimuli by probability summation among narrowband detectors," Vision Research. **17** , 37-652 (1977).

13. J. G. Robson and N. Graham. "Probability summation and regional variation in contrast sensitivity across the visual field," Vision Research. **21**, 409-418 (1981).

14. A. B. Watson. "Probability summation over time," Vision Research. **19**, 515-522 (1979).

15. N. Graham and J. Nachmias. "Detection of grating patterns containing two spatial frequencies: a comparison of single-channel and multiple-channel models," Vision Research. **11**, 251-259 (1971).

16. N. Graham, J. G. Robson and J. Nachmias. "Grating summation in fovea and periphery," Vision Research. **18**, 815-825 (1978).

17. A. B. Watson and J. Nachmias. "Summation of asynchronous gratings," Vision Research. **20** , 91-94 (1980).

18. A. Weber. *Image data base (USCIPI Report 1070)* (Image Processing Institute, University of Southern California, Los Angeles, CA, 1983).

19. A. B. Watson. "Receptive fields and visual representations," SPIE Proceedings. **1077**, 190-197 (1989).

DCT BASIS FUNCTION VISIBILITY: EFFECTS OF VIEWING DISTANCE AND CONTRAST MASKING

Andrew B. Watson Joshua A. Solomon Albert Ahumada

MS 262-2, NASA Ames Research Center, Moffett Field, CA 94035-1000
beau@vision.arc.nasa.gov al@vision.arc.nasa.gov jsolomon@vision.arc.nasa.gov

Alan Gale

San Jose State University

ABSTRACT

Several recent image compression standards rely upon the Discrete Cosine Transform (DCT). Models of DCT basis function visibility can be used to design quantization matrices for arbitrary viewing conditions and images. Here we report new results on the effects of viewing distance and contrast masking on basis function visibility. We measured contrast detection thresholds for DCT basis functions at viewing distances yielding 16, 32, and 64 pixels/degree. Our detection model has been elaborated to incorporate the observed effects. We have also measured detection thresholds for individual basis functions when superimposed upon another basis function of the same or a different frequency. We find considerable masking between nearby DCT frequencies. A model for these masking effects will also be presented.

1. INTRODUCTION

The JPEG, MPEG, and CCITT H.261 image compression standards, and several proposed HDTV schemes employ the Discrete Cosine Transform (DCT) as a basic mechanism [1, 2]. Typically the DCT is applied to 8 by 8 pixel blocks, followed by uniform quantization of the DCT coefficient matrix. The quantization bin-widths for the various coefficients are specified by a quantization matrix (QM). The QM is not defined by the standards, but is supplied by the user and stored or transmitted with the compressed images.

The principle that should guide the design of a QM is that it provide optimum visual quality for a given bit rate. QM design thus depends upon the visibility of quantization errors at the various DCT frequencies. In recent papers[3, 4], Peterson *et al.* have provided measurements of threshold amplitudes for DCT basis functions at one viewing distance and several mean luminances. Ahumada and Peterson [5] have devised a model that generalizes these measurements to other luminances and viewing distances, and Peterson *et al.* [6] have extended this model to deal with color images. From this model, a matrix can be computed which will insure that all quantization errors are below threshold. Watson [7] has shown how this model may be used to optimize the quantization matrix for an individual image.

2. EFFECTS OF DISPLAY RESOLUTION

Visual resolution of the display (in pixels/degree of visual angle) may be expected to have a strong effect upon the visibility of DCT basis functions, and we therefore collected data to document this effect and to validate and enhance the model.

0-8194-1474-3/94/$6.00

2.1 Practical Pixel Sizes

Visual resolution of the display (in pixels/degree of visual angle) is determined by display resolution (in pixels/cm) and viewing distance (in cm), according to the formula

$$(\text{pixels/degree}) = (\text{pixels/cm}) / \cot^{-1}[\text{distance}]$$

In the viewing situations for which block-DCT compression is contemplated, there are limits to the practical range of visual resolutions. At the high end, display resolution will be wasted on spatial frequencies which are not visible to the human eye. The limit of human spatial resolution is about 60 cycles/degree. Nyquist sampling of this frequency would require 120 pixels/degree. This corresponds to 300 dpi printing viewed at a distance of about 23 inches. At the low end, the pixel raster becomes visible. In these experiments, we have examined three viewing distances, 16, 32, and 64 pixels/degree, that span a large part of the range of useful viewing distances.

2.2 Methods

Detection thresholds for single basis functions were measured by a two-alternative, forced-choice method. Each trial consisted of two time intervals, within one of which the stimulus appeared. The stimulus was a single DCT basis function, added to the uniform gray background that remained throughout the experiment. Background luminance was 40 cd m^{-2}, and frame rate was 60 Hz. Observers viewed the display screen from distances of 48.7, 97.4, 194.8 cm. Display resolution was 37.65 pixels/cm. Images were magnified by two in each dimension, by pixel replication, to reduce monitor bandwidth limitations, resulting in magnified pixel sizes of 1/16, 1/32, and 1/64 of a degree, respectively at the three viewing distances (basis functions were 1/2, 1/4, and 1/8 degree in width). We describe these three viewing distances as yielding effective visual resolutions of 16, 32, and 64 (magnified) pixels/degree.

During presentation, the luminance contrast of the stimulus was a Gaussian function of time, with a duration of 32 frames (0.53 sec) between $e^{-\pi}$ points. The peak contrast on each trial was determined by an adaptive QUEST procedure [8], which converged to the contrast yielding 82% correct. After completion of 64 trials, thresholds were estimated by fitting a Weibull psychometric function [9]. Thresholds are expressed as contrast (peak luminance, less mean luminance, divided by mean luminance), converted to decibel sensitivities ($-20 \log_{10}[\text{threshold}]$)

To reduce the burden of data collection, we measured thresholds for only 30 of the possible 64 basis functions, as indicated in Fig. 1. To the extent that thresholds change slowly as a function of DCT frequency, this sampling constrains our model sufficiently.

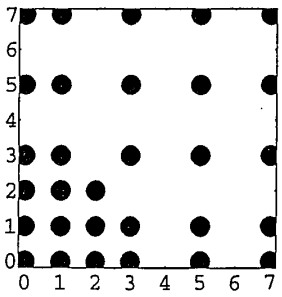


Figure 1. Subset of DCT frequencies used in the experiment.

To date, two data sets have been collected at the low resolution, five at the middle resolution, and one at the highest resolution, as shown in Table 1.

resolution (pixels/degree)	observer				
	abw	mjy	aig	sj	jas
16	0	30	0	30	0
32	7	30	60	30	30
64	0	30	0	2	0

Table 1. Thresholds collected for each observer and viewing distance.

2.3 Model of DCT Contrast Sensitivity

The model of DCT contrast sensitivity that we consider here is essentially that described by Peterson et al. [6] In that model, log sensitivity versus log frequency is a parabola, whose peak value, peak location, and width vary with mean luminance. In addition, sensitivity at oblique frequencies ($\{u\neq0, v\neq0\}$) is reduced by a factor that is attributed to the orientation tuning of visual channels. The parameters of significance here are s_0 (peak sensitivity), $f0$ (peak DCT frequency at high luminances), and k_0 (inverse of the *latus rectum* of the parabola), and r (the orientation effect).

2.4 Results

Figures 2, 3, and 4 show decibel contrast sensitivities for the three viewing distances, along with curves showing the predictions of the best fitting version of the model. Within each figure, the three panels show data for horizontal frequencies $\{u, 0\}$, vertical frequencies $\{0, v\}$, 45 degree orientations $\{u, v=u\}$, and the remaining obliques $\{u>0, 0<v\neq u\}$, all plotted against the radial frequency $f = \sqrt{u^2 + v^2}$. In the case of the obliques, because there is no simple one-dimensional prediction to plot, we plot instead the actual sensitivity minus that predicted by the model. These plots, and the fits, do not include the thresholds at {0,0} (DC), which are reserved for a separate discussion. The data at 64 pixels/degree also omit 3 thresholds at very high frequencies which we suspect to be artifactual.

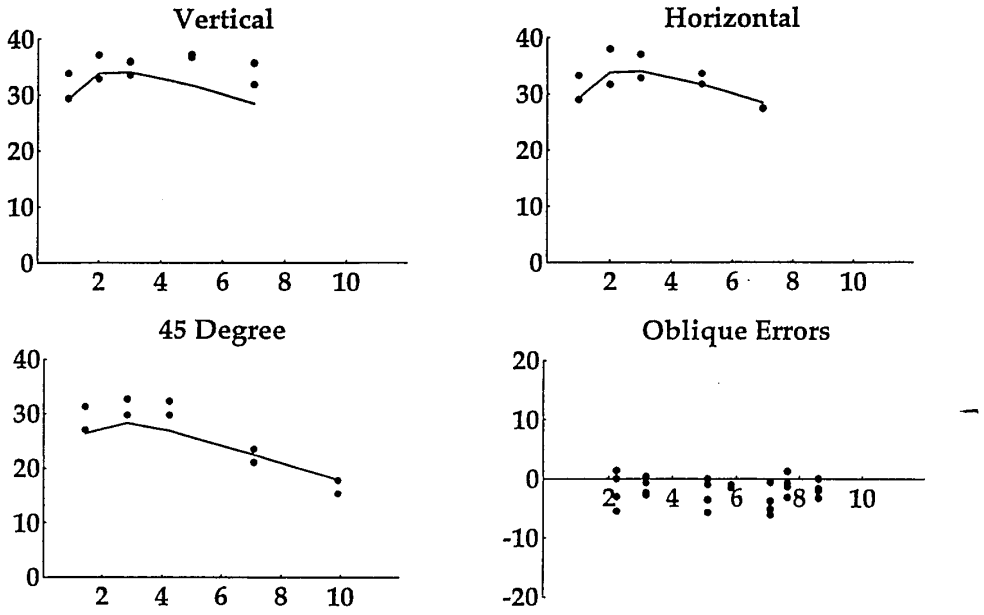


Figure 2. DCT basis function sensitivities at 16 pixels/degree.

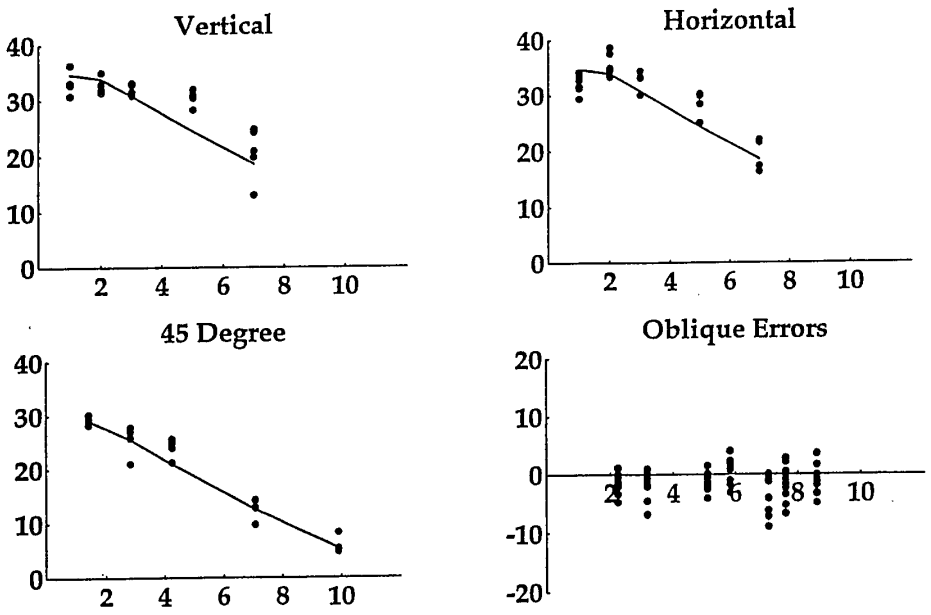


Figure 3. DCT basis function sensitivities at 32 pixels/degree.

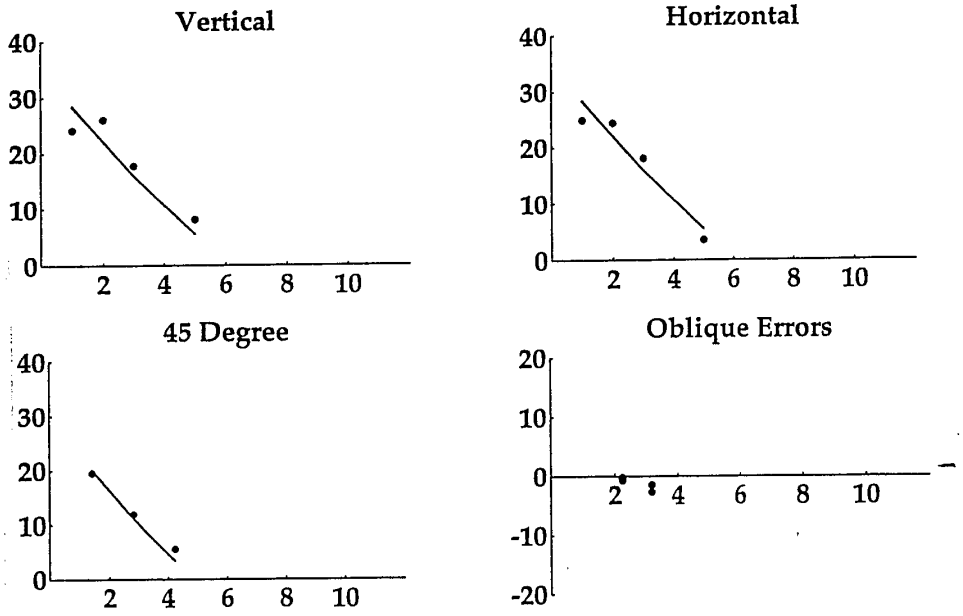


Figure 4. DCT basis function sensitivities at 64 pixels/degree.

The fits are reasonable, though there appear to be some systematic departures from the model. For reference, the RMS error of the raw data at the middle distance is 2.03 decibels, while the RMS error of the fit in Fig.s 2-4 is 2.94 decibels. The estimated parameters are shown in Table 2.

	pixels/degree		
	16	32	64
s0	51.1	56.17	29.84
f0	1.728		
k0	3.68		
r	0.5115		

Table 2. Estimated model parameters.

The parameters *f0, k0, and r* (related to peak frequency, bandwidth, and orientation effects) are equated for all resolutions, while a separate value of *s0* (peak contrast sensitivity) is estimated for each of the three resolutions. The behavior of this parameter is worth considering. Between 64 and 32 pixels/degree, it increases by a factor of 1.88. Between these two resolutions, the basis functions increase in size by a factor of two in each dimension. Thus if sensitivity increased linearly with area (as it should for very small targets [10, 11, 12]) we would expect an increase of a factor of 4. If sensitivity increased due only to spatial probability summation [13, 14], we would expect a factor of about $4^{1/4} = 1.414$. Thus the obtained effect is nearer to that expected of probability summation. At the closest viewing distance, despite a further magnification by 2, the parameter *s0* actual declines. While we would expect a smaller effect of size at the largest sizes, this decline is unexpected and may be due to 1) the relatively poor fit at this resolution, and 2) aspects of visual sensitivity which are not yet captured by the model.

2.5 DC Sensitivities

Figure 5 shows the sensitivities for DC basis functions at the three visual resolutions.

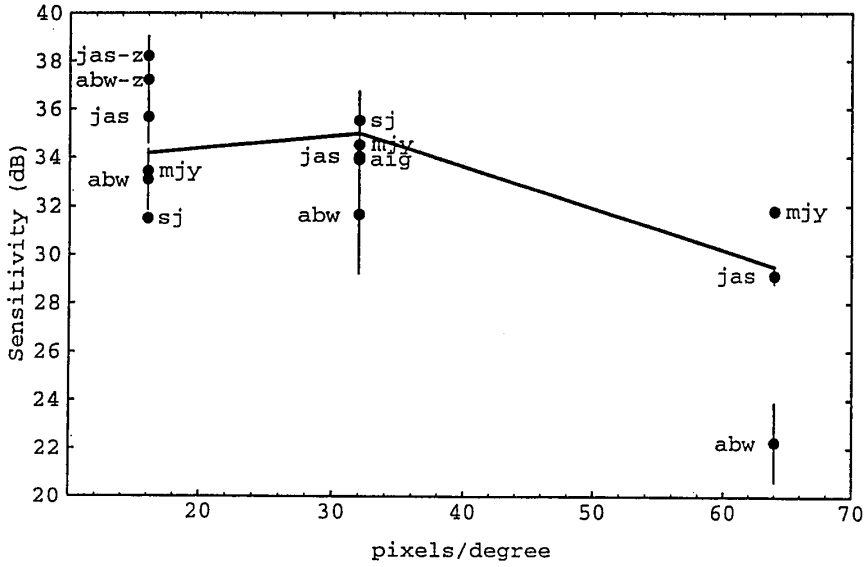


Figure 5. DC basis function sensitivities as a function of display visual resolution. Error bars of plus and minus one standard deviation are shown when multiple measurements were available. For clarity, points with error bars are labeled on the left, those without, on the right. The line indicates the parameter s0 from Table 2.

Ahumada et al. [5, 6] proposed as a working hypothesis that DC sensitivity is given by the peak sensitivity *s0*. This prediction is given by the line drawn in Fig. 5. It captures some of the variation in the DC sensitivities, but further data will be needed to adequately test this model. The points in Fig. 5 at a resolution of 16 pixels/degree and labeled with the suffix "-z" were obtained by pixel-replication at the middle viewing distance, rather than use of the near distance. Their enhanced sensitivity suggests that viewing distance per se may have an effect, even when visual resolution is held constant. The substantial variability of DC thresholds at the highest resolution may be due to differences in accommodation between observers.

2.6 Discussion

We have examined the variation in visibility of single DCT basis functions as a function of display visual resolution. We have shown that the existing model [5, 6] accommodates resolutions of 16, 32, and 64 pixels/degree, provided that one parameter, the peak sensitivity s0, is allowed to vary. Variations in this parameter are to some extent consistent with spatial summation, although sensitivity is lower at the lowest resolution than summation would predict.

Practical DCT quantization matrices must take into account both the visibility of single basis functions, and the spatial pooling of artifacts from block to block. Elsewhere we have shown that to a first approximation this pooling is consistent with probability summation[15]. If we consider two images of equivalent size in degrees, but visual resolutions differing by a factor of two, then the sensitivity to individual artifacts would be lower by $4^{1/4}$ in the higher resolution image due to the smaller block size in degrees, but higher by $4^{1/4}$ in the same image due to the greater number of blocks. Thus the same matrix should be used with both. The point of this example is that the overall gain of the best quantization matrix must take into account both display resolution and image size.

3. EFFECTS OF CONTRAST MASKING

3.1 Contrast masking

Watson[7] noted several image-dependent factors influencing the detectability of DCT basis functions and showed how to compute custom QMs for given images, in accord with these factors. One image-dependent factor influencing the detectability of DCT basis functions is contrast masking. Typically, sensitivity to quantization error, in a particular DCT coefficient, decreases with the magnitude of that coefficient. Watson's quantization scheme relies on the following model (based on work by Legge and Foley [16, 17]) for contrast masking: given a DCT coefficient c_T and a corresponding absolute threshold t_T, the masked threshold m_T will be

$$m_T = t_T \, \mathrm{Max}\left[1, \left| c_T / t_T \right|^{w_T} \right], \tag{1}$$

where w_T is an exponent that lies between 0 and 1. In the sequel, we will refer to this model as Model 1 In Model 1, sensitivity to a particular coefficient's quantization error is independent of the magnitudes of all the other coefficients (except the DC). Here we present data which indicate that sensitivity to a particular coefficient's quantization error is affected by the magnitudes of other coefficients. We propose a revision of Model 1 to account for between-coefficient contrast masking.

3.2 Methods

General methods were the same as in the earlier experiments (Section 2.2). Each stimulus was the sum of a test basis function and a mask basis function, added to the mean luminance of the display. The contrast of the mask remained constant throughout a block of 64 trials, while the contrast of the test was varied using the Quest procedure[8] to determine the threshold for the test in the presence of the mask. Effective visual resolution was 32 pixels/degree, so that each stimulus subtended 0.25 degrees by 0.25 degrees.

Masked thresholds m_T for four test DCT frequencies were measured as a function of masking contrast for three different mask frequencies. The tests frequencies $\mathbf{T}$ were {0,0}, {0,1}, {0,3} and {0,7}. These last three also served as the masks. Additionally, {1,1} and {1,0} were used to mask {0,1}; and {2,2} was used to mask {0,3}. Un-masked threshold t_T was also determined for each test. Theoretically, DCT coefficients can assume any real value. In the current study we use coefficients c_U, such that $0 \le c_U \le 1$. A coefficient with value 1 fully utilizes the dynamic range of the display. For nearly every test/mask combination, six masking contrasts were used. Here we express these contrasts in decibels ($dB[c_U] = 20\ \log_{10}[c_U]$): -36, -30, -24, -18, -12 and -6. Because t_{07} is so high, when this basis function served to mask others, only the four greatest masking contrasts were used. Test and mask frequencies were fixed within a block of trials, and frequency combinations were run in a randomized fashion. The second author (jas) was the only observer in these experiments.

3.3 Results and Discussion

The results are plotted in Figs. 6 and 7.

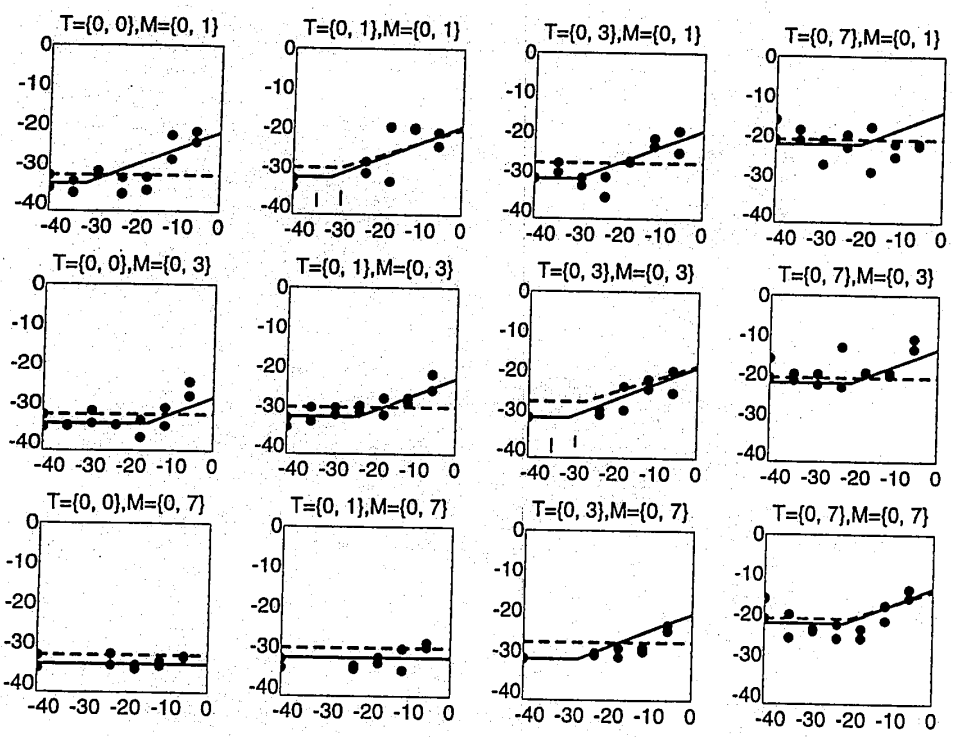


Fig. 6. Masked thresholds ($dB[m_T]$) for four test basis functions are plotted as a function of masking contrast ($dB[c_M]$) for three different masks. Unmasked thresholds ($dB[t_T]$) for the test basis functions are plotted on the ordinates. The dashed and solid lines are the predictions of Models 1 and 2, respectively, as described in the text.

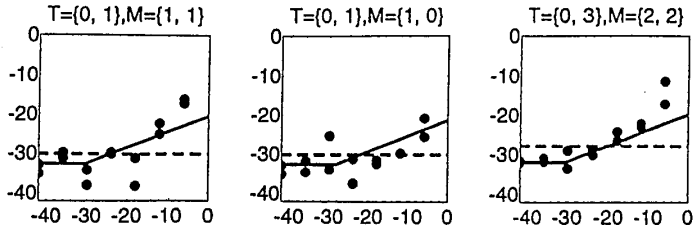


Fig. 7. Masked thresholds for test {0,1} as a function of masking contrast for the masks {1,1} and {1,0}, and for test {0,3} as a function of masking contrast for the mask {2,2}.

3.3.1 The dipper effect

Data gathered with the {0,1}/{0,1} test/mask combination at masking contrasts of -36 and -30 dB have been omitted from further analysis. Similarly, we have omitted the {0,3}/{0,3} data at -36 and -30 dB. These data appear as short vertical line segments in Fig. 6. Measured thresholds for these four viewing conditions fall well below their corresponding unmasked thresholds. These data demonstrate the "dipper effect," a well-documented phenomenon wherein a low contrast grating increases the detectability of a grating of the same frequency and phase[16, 18, 19]. These data have been omitted because it is not clear that the dipper effect comes into play for natural images. For images composed of more than one 8x8 pixel block, DCT basis functions can appear as gratings (uniform values) or noise (random values; with a quantifiable variance) or anything in between. The dipper effect would appear if both test and mask were gratings. However, there is no indication that it would appear otherwise. The influence of a particular DCT coefficient on the detectability of quantization errors in natural images is similar in concept to the influence of a grating on the detectability of random visual noise. No dipper effect is expected in such a paradigm. Since we ultimately wish to model the detectability of quantization error in natural images, we believe that the exclusion of the "dipper data" will benefit our initial approximations.

3.3.2 Model 1

Model 1 was fit to the data. Model 1 does not include between-coefficient contrast masking. Consequently, for any given test basis function, its prediction for masked threshold is the same constant function of masking contrast for every mask having a non-zero coefficient at a different DCT index than the test. By setting all of the w_Ts in Eq. 1. equal to a single parameter w, the total variance (on a log scale) from the model increased by less than 0.3%. Hereafter, when we refer to Model 1, we mean specifically: Given a test DCT basis function c_T, its corresponding absolute threshold t_T and a mask DCT basis function c_M the masked threshold m_T will be

$$m_T = \begin{cases} t_T \, \mathrm{Max}\left[1, (c_M/t_T)^w\right] & \text{for} \quad T = M \\ t_T & \text{otherwise} \end{cases}, \tag{2}$$

where $0 \le w \le 1$. Best fitting (method of least squares) values for w and t_T, as determined for Model 1, are given in Table 3. For comparison, we have also analyzed a Model 0 which predicts no contrast masking, i.e. $m_T = t_T \forall T$. Best fitting values for t_T, as determined by Model 0 are also given in Table 3. Model 1 reflects the data for the viewing conditions in which the mask and target were identical more accurately than Model 0 does. However, it cannot reflect the between-coefficient masking evident by the increase in measured threshold with masking contrast for the other test/mask combinations.

3.3.3 Model 2

In order to reflect the between-coefficient masking, we propose the following revision of Model 1, referred to hereafter as Model 2. Given a test DCT basis function c_T, its corresponding absolute threshold t_T and a mask DCT basis function c_M the masked threshold m_T will be

$$m_T = t_T \, \mathrm{Max}\left[1 , \left(f[T,M] \frac{c_M}{t_T} \right)^w \right], \tag{3}$$

where w is an exponent that lies between 0 and 1 and $f[T,M]$ is a positive, frequency-dependent scaling factor, that assumes a maximum value of 1 when $T = M$. $f[T,M]$ may be described as a family of tuning functions. That is, for any test basis function c_T, $f[T,M]$ reflects the sensitivity of c_T detection to masks at different frequencies. We have chosen to specify these sensitivity functions with the following one-parameter rule:

$$f[T,M] = \exp\left[-\pi \|T - M\|^2 / \varsigma_T^2 \right], \tag{4}$$

where $\varsigma_T = \varsigma \, \mathrm{Max}[1, \|T\|]$.. This is a radially symmetric Gaussian sensitivity function with a bandwidth that increases in proportion to frequency (except at DC). This is analogous to the spatial frequency channels that are believed to underlie the early stages of human visual processing.

Best fitting (method of least squares) values for ς, w and t_T, as determined for Model 2, are also given in Table 3. The average variance (squared rms error on a decibel scale) from Models 0, 1 and 2 is also provided in Table 3. The best fitting predictions of Model 2 are also drawn as solid lines in Figs. 6 and 7.

Parameter	Model 0	Model 1	Model 2
$\mathrm{dB}[t_{00}]$	-32.9	-32.9	-35.1
$\mathrm{dB}[t_{01}]$	-29.2	-30.2	-32.6
$\mathrm{dB}[t_{03}]$	-27.4	-27.8	-31.9
$\mathrm{dB}[t_{07}]$	-20.5	-20.9	-22.1
w	n/a	0.324	0.396
ς	n/a	n/a	5.5
Average variance from model	18.5	16.5	8.95

Table 3. Residual variance from Models 0, 1 and 2.

3.4 Conclusions

With the addition of a single parameter (ς), our Model 1 captures 46% more of the variance in our data than does Model 0. Incorporating this modification into the current method for computing DCT quantization matrices will yield more efficient image compression. The estimated value of ς indicates a rather broad bandwidth for the masking effect. This may be due in part to the rather broad bandwidth of the basis functions themselves.

4. ACKNOWLEDGMENTS

We thank Mark Young for extensive assistance and Heidi Peterson for useful discussions. This work was supported by NASA RTOPs 506-59-65 and 505-64-53.

5. REFERENCES

1. W.B. Pennebaker and J.L. Mitchell,JPEG Still image data compression standard, Van Nostrand Reinhold, New York (1993).
2. G. Wallace,"The JPEG still picture compression standard," Communications of the ACM, 34(4), 30-44 (1991).
3. H.A. Peterson,"DCT basis function visibility in RGB space," (1992).

4. H.A. Peterson, H. Peng, J.H. Morgan and W.B. Pennebaker,"Quantization of color image components in the DCT domain," (1991).

5. A.J. Ahumada Jr. and H.A. Peterson,"Luminance-Model-Based DCT Quantization for Color Image Compression," (1992).

6. H. Peterson, A. Ahumada and A. Watson,"An Improved Detection Model for DCT Coefficient Quantization," (1993).

7. A.B. Watson,"DCT quantization matrices visually optimized for individual images," (1993).

8. A.B. Watson and D.G. Pelli,"QUEST: A Bayesian adaptive psychometric method," Perception and Psychophysics, 33(2), 113-120 (1983).

9. A.B. Watson,"Probability summation over time," Vision Research, 19, 515-522 (1979).

10. C. Noorlander, M.J.G. Heuts and J.J. Koenderink,"Influence of the target size on the detection threshold for luminance and chromaticity contrast," Journal of the Optical Society of America, 70(9), 1116-1121 (1980).

11. C.H. Graham, R.H. Brown and F.A. Mote,"The relation of size of stimulus and intensity in the human eye: I. Intensity thresholds for white light," J. Exp. Psychol., 24, 555-573 (1939).

12. H.B. Barlow,"Temporal and spatial summation in human vision at different background intensities," Journal of Physiology, 141 , 337-350 (1958).

13. N. Graham, J.G. Robson and J. Nachmias,"Grating summation in fovea and periphery," Vision Research, 18 , 815-825 (1978).

14. J.G. Robson and N. Graham,"Probability summation and regional variation in contrast sensitivity across the visual field," Vision Research, 21, 409-418 (1981).

15. H.A. Peterson, A.J. Ahumada Jr. and A.B. Watson,"The Visibility of DCT Quantization Noise," SID Digest of Technical Papers, XXIV, 942-945 (1993).

16. G.E. Legge and J.M. Foley,"Contrast masking in human vision," Journal of the Optical Society of America, 70(12), 1458-1471 (1980).

17. G.E. Legge,"A power law for contrast discrimination," Vision Research, 21, 457-467 (1981).

18. C.F. Stromeyer III and S. Klein,"Spatial frequency channels in human vision as asymmetric (edge) mechanisms," Vision Research, 14, 1409- 1420 (1974).

19. J. Nachmias and R. Sansbury,"Grating contrast: discrimination may be better than detection," Vision Research, 14 , 1039-1042 (1974).