CCITT Study Group XV
Working Party XV/1
Specialists Group on Coding for Visual Telephony

Doc #121
June 1986

Source: UK, Sweden, Netherlands

Title : Accuracy of Transform Arithmetic

INTRODUCTION

There are two reasons why the accuracy of the transform computation is important:

Objective #1. Absolute accuracy. To ensure that the energy packing and decorrelation functions are sufficiently well carried out so as to give efficient coding.

Objective #2. Relative accuracy. To ensure that all implementations of the inverse transform give identical results otherwise the pictures in the coding and decoding loops differ, which has important consequences in any inter-picture predictive scheme.

The first objective is difficult to quantify at this stage and is part of the reason for choosing the classical matrix multiplier approach for the flexible hardware. It may also depend on the transform which is finally adopted.

INITIAL APPROACH and its FAILURE

Initially, an attempt was made to meet these two objectives by the following reasoning. The inverse quantizer had already been specified as having up to 12 bits out. The inverse transform should give 9 bits out as the decoded prediction error in inter-picture coding mode. The DCT was a likely transform. Therefore, it was thought that the accuracy of the inverse transform hardware should be sufficient that it would give the same result as as an infinite precision DCT operating with 12 bit data input and rounded to 9 bits at the end. With this definition it was not necessary to specify the quantization accuracy of the matrix elements or how many bits had to be preserved between the two 1-D transforms.

Two problems arose. The first was how to be sure that a particular design did meet the specification. The second is more fundamental. Although by working to more digits internally the error introduced by the hardware can be made smaller and smaller, there can always be occasions when the hardware result and the infinite precision value round to integers which differ by one.

NEW APPROACH

The above means that the only way to guarantee hardware compatibility is to specify the hardware architecture and the number of bits at each stage with the rules for discarding bits. This meets objective #2 by definition.

Objective #1 cannot be resolved yet but the following is proposed. Specify hardware which will give the maximum accuracy we might require, provided that the complexity of such hardware is acceptable. We cannot do better than an occasional error of 1 in ±255 compared to the rounded precise DCT so that is a suitable target. It is better to have precision in the flexible hardware that we might eventually not require than have to do a major redesign later.

SIMULATION of HARDWARE

Calculations have been performed to determine the effect of the number of bits used to represent the matrix elements and the intermediate values carried between the two 1-D parts making up the 2-D transform or the inverse. The multiplier and accumulator were assumed to have sufficient capacity that no bits were lost internally.

A pel domain test block was generated by uniform random numbers:

-255	107	-83	6	5	-192	98	-77
66	-76	-158	-140	104	213	-240	-153
-221	-8	-38	140	-38	111	198	-79
130	137	233	-72	-23	218	194	-48
-40	-11	-179	180	3	-181	-6	-242
-184	-203	-54	-53	-52	149	68	-192
210	98	-190	-82	174	164	-195	-238
-81	21	121	-20	45	-141	229	32

Two 1-D DCTs were applied to this pel domain block and rounded to 12 bit integers (±2048). Computation was by Microsoft Pascal with REAL8 precision. The resulting transfom block was:

-99	-10	-225	246	-200	48	-173	-7
-51	-69	-30	-63	-46	-59	-28	-94
-77	-25	51	-61	85	-182	-76	98
-300	47	-93	68	111	-29	-79	-55
126	45	126	-349	-56	106	-240	157
201	66	76	48	-150	-63	6	-2
-34	-341	-70	-357	-200	224	-166	43
-118	69	-101	-63	188	27	-299	-120

PEL to TRANSFORM to PEL

The overall errors were computed for pel domain through the transform domain and back to pel domain. The following parameters apply to the table below:

c = number of bits, including a sign bit, used to represent the matrix elements in all four 1-D DCTs. These c bits covered the range -0.5 to +0.49999.....

b = number of bits, including a sign bit, carried forward by truncation, between the two 1-D operations in each of the forward and inverse DCTs. Of these b bits, 11 are for the sign and integer part. (Maximum range from DCT is ±721.25....)

12 = number of bits, including a sign bit, carried by rounding, between the forward DCT and the inverse DCT. (This is equivalent to integer rounding as maximum range for the DCT is ±2040.)

N = number of pels, out of 64, where the simulated hardware result differed from the input pel of the test block.

e = magnitude of the largest error in the above N pels.

c	b	N	e
8	11	44	3
8	12	45	3
8	13	40	3
8	14	43	4
8	15	40	4
8	16	46	4
10	11	45	2
10	12	20	1
10	13	9	1
10	14	4	1
10	15	5	1
10	16	4	1

12	11	43	2
12	12	21	1
12	13	10	1
12	14	4	1
12	15	4	1
12	16	4	1
14	11	39	2
14	12	25	1
14	13	6	1
14	14	6	1
14	15	3	1
14	16	3	1
16	11	39	2
16	12	24	1
16	13	8	1
16	14	6	1
16	15	3	1
16	16	2	1

TRANSFORM to PEL

The table below is for the inverse transform only, operating on the transfrom domain values derived above. Errors are between 9 bit rounded (±255) versions of the hardware simulator and Pascal REAL8 inverse DCT results.

c	b	N	e
8	11	30	2
8	12	30	2
8	13	33	2
8	14	38	2
8	15	37	2
8	16	38	2
10	11	29	1
10	12	17	1
10	13	9	1
10	14	7	1
10	15	6	1
10	16	7	1
12	11	30	2
12	12	15	1
12	13	7	1
12	14	6	1
12	15	4	1
12	16	1	1
14	11	26	2
14	12	13	1
14	13	6	1
14	14	7	1
14	15	4	1
14	16	1	1
16	11	26	2
16	12	14	1
16	13	7	1
16	14	6	1
16	15	4	1
16	16	2	1

DISCUSSION of RESULTS

The above tables are for only one set of input data. Therefore the results should be treated with some caution - especially the differences caused by adjacent values of b. However, general trends can be seen. Use of 16 by 16 Multiplier Accumulators is a safe path for the flexible prototype hardware. 12 by 12 might be sufficient. Matrix elements held as 7 bits plus sign are not adequate.

The above simulations took advantage of the fact that the matrix elements for the DCT do not exceed the range -0.5 to +0.4999... For other transforms this may not be the case. In particular, if the value of +1 is needed, then 2 more bits are required. From a hardware viewpoint, this means that either the PROMs hold absolute values in the range -2 to +1.99999.. or they hold scaled versions of the elements (with the second bit set to '1') and arrangements are made for selecting the appropriate shifted bits at the Multiplier Accumulator output. If the former case, then for 16 bit coefficient PROMs a value of c=14 is applicable in the above tables.

Looking at available devices it is found that the TRW family and equivalents is dominant. The 12 by 12 and 16 by 16 are available at very similar prices and they are both in the same package size. Therefore the TRW TDC1010 architecture which is available from several sources and in various technologies would be a good choice for the flexible hardware.

CONCLUSION

The matrix multiplier implementation of the inverse transform should use TDC1010 or functionally equivalent devices. The hardware should allow matrix coefficients to be stored to 16 bits. The most significant 16 bits by truncation should be carried between the two 1-D transforms. The same is suggested for the forward transform though it is unnecessary to specify this to achieve compatibility. The output of the inverse transform should be rounded to 9 bits giving pel domain values in the range -256 to 255.