INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG 98/M3832
July 1998

Source: University of California at Berkeley
Purpose: Proposal
Title: Changes regarding Matching Pursuits in Video VM V.11
Authors: Sen-ching Samson Cheung, Ralph Neff, Avideh Zakhor[1]

The first section describes proposed changes in the "Matching Pursuit inter texture coding mode" section of Video Verification Model Version 11.0 (N2172). The second section contains results produced with all the new changes incorporated in the codec.

1. Changes to VM

1.1 Decoder Syntax Change

This section contains changes that affect decoding operations and bitstream syntax.

1.1.1 Intra Block Coding

1.1.1.1 Summary

In the previous versions, all the DC values for intra coded macroblocks were coded with 5-bit fixed length codes. We propose to adopt the same intra DC prediction and coding method used in DCT intra coding. As high fidelity details will be coded using Matching Pursuit atoms, the quantization parameter shall be set permanently to 19 in coding the DC values.

1.1.1.2 Text change

1. In 14.3.3 "Motion Compensation", second paragraph, fourth sentence:

Old text :

The DC level of each of the six subblocks (four luma and two chroma) is quantized to five bits and transmitted to the decoder.

New text:

The DC level of each of the six blocks are coded in exactly the same way as the intra dc coefficient of DCT described in section 3.4.6. and 3.4.7 using intra DC VLC with short_video_header being 0 and quant_type being 1. The quantization parameter used is always set to 19. The decoded dc coefficients shall be scaled down by a factor of 8 with rounding

2. In 14.3.7.3 "Motion Shape Texture Syntax", second paragraph, third sentence:

Old text:

…For INTRA and INTRARefresh macroblocks, the DC intra coefficients for each sub-block are quantized to 5 bits and transmitted without prediction.

New text:

[1] For more information on this document, please contact Dr. Avideh Zakhor at avz@eecs.berkeley.edu.

…For INTRA and INTRARefresh macroblocks, the DC intra coefficients for each block are transmitted using dct_dc_size_luminance and dct_dc_differential with identical definitions as their counterparts in DCT. The actual dc value used is obtained by scaling down the decoded dc coefficient by 8 with rounding.

1.1.2 Integer Matching Pursuit Reconstruction

1.1.2.1 Summary

Floating point specifications in Matching Pursuit library and atom pasting have been replaced by integer ones in VM V10.0 to reduce complexity and prevent accumulation of possible mismatch error from intra vop. New provisions are added this time to limit the size of intermediate results while maintaining an order independence property -- atoms can be added in any order.

1.1.2.2 Text Change

1. In section 14.3.5 "Prediction Error Reconstruction", first paragraph, first sentence:

Old text:

The reconstruction of prediction error can be done entirely in integer with 16-bit precision.

New text:

Ten-bit precision will be sufficient in constructing the reconstruction VOP.

2. In section 14.3.5 "Prediction Error Reconstruction", last sentence of second paragraph:

Old text:

… The range of s[y][x] is [-1865:1865].

All the reconstructed atoms described above are pasted together to form a complete prediction error vop. Luminance mean adjustment is also added back if available. Any intermediate sum shall be clipped between [-32767:32767]. The resulting prediction error shall be clipped between [-255:255]. The last step is to add the prediction error back to the motion compensated prediction and clip the final result between [0:255].

New text:

All the reconstructed atoms described above as well as the luminance mean adjustment (if available) are added back to the motion compensated prediction. The summation uses a special modulo-1024 arithmetic which is centered on the normal range [0:255]. In such an arithmetic, if an intermediate summation results a *falls beyond the range of [-384:639], it shall be mapped to* b *inside [-384:639] where* $a = k \times 1024 + b$ *with* k *being an integer. After the summation is complete, the final result is clipped between [0:255]. The purpose of having the modulo arithmetic is to limit the dynamic range of intermediate results. As modulo arithmetic is distributive, it can be applied to the intermediate results during the summation to ensure that the dynamic range will not exceed [-384:639]. It should be noted that a different modulo can be applied to intermediate results as long as :*

1. *the modulo used must be larger than 1024 and the principle range is centered around [0:255], and*
2. *the modulo-1024 arithmetic described above is done in the very last step before clipping.*

The final results will be identical to the case when modulo 1024 is used throughout the summation. The purpose of using a different modulo is to reduce the number of operations for modulo arithmetic by taking advantage of the longer word length in modern CPU.

1.1.3 Other Changes

1. In 14.3.4.1 "The Dictionary Set", the formula for a discrete 1-D Gabor function of size N is wrong :

Old Text:

$$g_{\vec{a}}(i) = K_{\vec{a}} \cdot g\left(\frac{i - N/2 + 1}{s}\right) \cdot \cos\left(\frac{2px(i - N/2 + 1)}{N} + f\right), \qquad i \in \{0,1,\dots N-1\};$$

New Text:

$$g_{\vec{a}}(i) = K_{\vec{a}} \cdot g(\frac{i - N/2 + 1}{s}) \cdot \cos(\frac{2px(i-(N-1)/2)}{16} + f), \qquad i \in \{0,1,\ldots N-1\};$$

2. In 14.3.4.1, "The Dictionary Set", Table "Parameter triples defining the 2-D separable Gabor Basis Dictionary", row 19:

Old text:

4 4 0 7

New text:

4 4 0 5

3. In 14.3.4.4 ,"Detail Description of Find Atom", first paragraph, seventh sentence:

Old text:

A fixed-stepsize linear quantizer with a quantization stepsize of 30 is used.

New text:

A fixed-stepsize linear quantizer with a quantization stepsize of 32 is used.

4. In 14.3.4.4, "Detail Description of Find Atom", first paragraph, the fourth line from the end:

Old text:

…In such cases, modulus of subsequent atoms can be quantized more finely around zero for up to three reconstruction levels: 24, 12, and 6.

New text:

…In such cases, modulus of subsequent atoms can be quantized more finely around zero for up to three reconstruction levels: 12, 6 and 3.

5. In 14.3.7.3, "Motion Shape Texture Syntax", syntax table for mp_atom:

Old text:

Modulus_flc 5 uimsbf

New text:

Modulus_flc 8 uimsbf

6. In 14.3.7.3 "Motion Shape Texture Syntax", fourth paragraph, seventh sentence:

Old text:

…An escape code followed by a 5-bit fixed-length modulus_flc (ranging from 0 to 26) extends the range of the index up to 57.

New text:

…An escape code followed by a 8-bit fixed-length modulus_flc (ranging from 0 to 255) extends the range of the index up to 286.

1.2 Encoder Optimizations

This section contains changes that affect encoding decisions only.

1.2.1 Motion Techniques

1.2.1.1 Summary

A number of simple changes to motion estimation and mode decision are recommended to better suit Matching Pursuit residual coding. These changes are mostly targeted at artifacts removal.

1.2.1.2 Text Change

New section 14.3.3:

14.3.3 Motion Estimation and Mode Decision

Motion Estimation and mode decision logic are similar to that described in section 3.3.2 except for the following changes:

1. *The Intra/Inter decision differs from that in 3.3.2.3 by also using dynamic ranges of the difference block and intra block in the decision. The idea is that the high frequency components in the inter block which are results of either failure of motion model and/or artifacts from previous reconstructed VOP, are visually less pleasing than a coarse approximation produced by an intra block. The amount of high frequency components of the inter-block is loosely measured by the maximum dynamic range* DR_{inter}*, that is the difference between maximum and minimum pixel values, of all the four 8x8 luma difference blocks. The dynamic range for the intra-block* DR_{intra} *is also computed. The new intra/inter mode decision logic is depicted as follows:*

   if ((DR_{inter} *>= (35 - MB_mean/7)) && (A < (*SAD_{inter} *- (*DR_{Intra}/DR_{Inter} *- 1)**N_B*)))*

   Use INTRA

   else

   Use INTER

2. *For integer pixel motion estimation, a simple technique is used to decide whether previous reconstructed VOP or previous original VOP is used in a macroblock by macroblock basis. Using previous reconstructed VOP allows better compensation of artifacts present in previous VOPs. On the other hand, better motion vectors, in terms of consistency and better texture mapping, are possible by using previous original VOP, especially in low bit rate coding. Without carrying out an exhaustive search on both reference VOPs, a simple prediction technique is used based on the previous residual VOP -- if the macroblock of the same spatial location from the previous residual is not present or intra-coded, previous original VOP is always used. Otherwise, the decision will be based on the Sum of Absolute Difference (SAD) and the Dynamic Range (DR) of the macroblock (luma only) from the previous residual using the following logic:*

   if ((SAD>1000) || (DR>40))

   Use previous reconstructed VOP

   else

   Use original reconstructed VOP

   The SAD is computed as the sum of the absolute values of all the residual pixels in the luma macroblock. As for the DR, if the boundary rectangle of the VOP is larger than or equal to CIF, it is just the difference between the largest and the smallest residual pixels in the luma macroblock. If the boundary rectangle of the VOP is smaller than CIF, the DR value used is the maximum of the dynamic range of the macroblock as well as those of the four 16x16 blocks displaced 8 pixels on each of the four directions : up, down, left and right. The above decision applies to both 16x16 vectors and 8x8 vectors but the half-pel estimation will still be based on previous reconstructed VOP.

3. *The overlapped block motion compensation forms an integral part of Matching Pursuit coding. It allows pixels extending beyond the current reference block to be used in mixing with predictions of neighboring blocks. However, the block matching motion estimation routine disregards this extended region and occasionally produces wrong prediction -- like casting dark edges at picture boundary inside the picture, for example. Without imposing a significant increase in complexities by employing overlapped block motion estimation, a simple technique, which decides between the zero vector and the best vector computed by the block matching, is used. The idea is based on the fact that zero vector is the most commonly used vector in low bit rate coding and it does not produce artifacts described above. This step is performed after the half-pel estimation is done. To incorporate the effect of the OBMC window, a weighted SAD function is computed over a 16x16 region centered on each of the 8x8 luma blocks:*

   $$WSAD_8(x,y) = \sum_{i=-3,\,j=-3}^{12,12} w(i,j) * |\, original - previous \,| * (!(Alpha_{original} == 0))$$

   and $$WSAD_{16}(x,y) = \sum_{1}^{K} WSAD_8(x,y)$$, where 0<K<=4 is the number of 8x8 blocks.

   w(i,j) is the corresponding values for the windowing function used in OBMC. For Matching Pursuit, the values of w(i,j) are shown below:

0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	1	2	3	4	5	5	5	5	4	3	2	1	0	0
0	1	3	6	8	11	13	14	14	13	11	8	6	3	1	0
0	2	6	10	15	20	24	26	26	24	20	15	10	6	2	0
0	3	8	15	24	30	35	38	38	35	30	24	15	8	3	0
0	4	11	20	30	39	46	50	50	46	39	30	20	11	4	0
1	5	13	24	35	46	54	58	58	54	46	35	24	13	5	1
1	5	14	26	38	50	58	62	62	58	50	38	26	15	5	1
1	5	14	26	38	50	58	62	62	58	50	38	26	15	5	1
1	5	13	24	35	46	54	58	58	54	46	35	24	13	5	1
0	4	11	20	30	39	46	50	50	46	39	30	20	11	4	0
0	3	8	15	24	30	35	38	38	35	30	24	15	8	3	0
0	2	6	10	15	20	24	26	26	24	20	15	10	6	2	0
0	1	3	6	8	11	13	14	14	13	11	8	6	3	1	0
0	0	1	2	3	4	5	5	5	5	4	3	2	1	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

The decision logic becomes:

if $(WSAD_{16}(best\_x,best\_y) > WSAD_{16}(0,0))$

Use zero vector

else

Use best motion vector

1.2.2 VOP Boundary Handling

1.2.2.1 Summary

A small gain in coding efficiency can be obtained during the atom search phase by rescaling the candidate atom extending beyond the boundary such that the norm of the overlapped portion of the basis with the VOP becomes one. This is entirely an encoder issue as the scaling factor will be accounted for in the modulus being coded.

1.2.2.2 Text Change

1. In add the following sentence after the second sence of the first paragraph:

New text:

One possible way to handle the cases when the basis functions extend beyond the VOP boundary is to first multiply the basis function by a scale factor such that the norm (square root of sum of square) of the portion of the basis function within the VOP becomes one.

2. In section 14.3.4.4, first paragraph, fourth sentence:

Old text:

The value of the largest inner product becomes the expansion coefficient, which much be quantized and transmitted to the decoder.

New text:

The value of the largest inner product (multiplied by the normalizing scale factor if this best basis function extends beyond the picture boundary) becomes the expansion coefficient, which much be quantized and transmitted to the decoder.

2. Simulation Results

The DCT_VM results are produced by the MoMuSys VM (VM8-971006.) The standard coding efficiency conditions are used in all cases. The Matching Pursuits (MP) runs use rate control attempting to match the bitrate of DCT_VM runs on a frame-by-frame basis.

Sequences	Mode	Bits	Y-PSNR	U-PSNR	V-PSNR
Container, qcif	DCT_VM	103063	29.89	37.00	36.54
7.5fps	MP	103530	31.46	37.46	37.49
10kbps	MP-DCT_VM		+1.57	+0.46	+0.95

Hall, qcif	DCT_VM	101387	30.32	36.52	39.62
7.5fps	MP	101351	31.77	36.49	39.62
10kbps	MP-DCT_VM		+1.45	-0.03	+0.00

Mother, qcif	DCT_VM	105795	32.66	38.73	39.65
7.5fps	MP	105805	33.26	38.84	39.80
10kbps	MP-DCT_VM		+0.60	+0.11	+0.15

Container, qcif	DCT_VM	261371	33.24	39.47	38.62
10fps	MP	261374	34.29	40.33	40.23
24kbps	MP-DCT_VM		+1.05	+0.86	+1.61

Silent, qcif	DCT_VM	253467	31.04	35.27	36.94
10fps	MP	253867	32.23	36.38	37.62
24kbps	MP-DCT_VM		+1.19	+1.11	+0.68

Mother, qcif	DCT_VM	235523	35.18	40.14	40.92
10fps	MP	235458	35.94	40.69	41.53
24kbps	MP-DCT_VM		+0.76	+0.55	+0.61

Coastgd, qcif	DCT_VM	515867	29.40	40.00	41.90
10fps	MP	515413	30.15	40.40	42.26
48kbps	MP-DCT_VM		+0.75	+0.40	+0.36

Foreman, qcif	DCT_VM	496019	31.13	37.22	37.39
10fps	MP	496229	31.69	38.22	38.69
48kbps	MP-DCT_VM		+0.53	+1.00	+1.30

News, cif	DCT_VM	479391	31.20	35.95	37.39
7.5fps	MP	480556	32.52	37.45	38.15
48kbps	MP-DCT_VM		+1.32	+1.50	+0.76

Coastguard, cif	DCT_VM	1318019	26.15	38.03	40.27
15fps	MP	1318120	27.20	37.79	41.07
112kbps	MP-DCT_VM		+1.05	-0.24	+0.80

Foreman, cif	DCT_VM	1380939	28.67	35.40	35.89
15fps	MP	1381076	30.01	36.53	37.35
112kbps	MP-DCT_VM		+1.34	+1.13	+1.46

News, cif	DCT_VM	1153763	34.19	38.26	39.08
15fps	MP	1158525	35.65	40.06	40.54
112kbps	MP-DCT_VM		+1.46	+1.80	+1.46

Mobile, cif	DCT_VM	10198875	26.45	30.83	30.36
30fps	MP	10197438	26.85	33.26	32.94
1Mbps	MP-DCT_VM		+0.40	+2.43	+2.58

Stefan, cif	DCT_VM	10640971	29.45	34.87	34.50
30fps	MP	10639566	29.79	36.76	36.63
1Mbps	MP-DCT_VM		+0.34	+1.89	+2.13