/0

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

Document #406
December 6, 1988

SOURCE: Japan
TITLE: Loop Filter Investigations in Japan

1. Introduction

This document summarizes loop filter investigations performed in the Japanese natinal committee. The studied items are the number and positions of filters(Annex 1), filtering method on block boundaries(Annex 2), picture qualities in high bit-rate operations (Annex 3) and loop filter robustness under IDCT mimatch condition (Annex 4).

2. Summary of Experimental Results

This section summarizes conclusions of Annex 1 - 4.

(1) Number and positions of filters (Annex1)

- By using an improved loop filter before frame memory, block-shape and mosquito noises are significantly reduced.
- A combination of RM6 type loop filter and an adaptive post filter gives better picture quality than only RM6 type loop filter but slightly worse picture quality than only the adaptive loop filter.
- From complexity and picture quality points of view, candidates for final decision can be: a) RM6 type loop filter plus an adaptive post filter and b) adaptive loop filter. Here, the adaptive filter should be placed before frame memory in order to use coding results as adaptation sources.

(2) Filtering method on block boundaries (Annex 2)

- For RM6 type filters, the picture quality difference between 8x8 and 10x10 is insignificant.
- Improved adaptive filters should be 10 x 10.
- Hardware complexity comparison depends on implementations.

(3) Picture qualities in high bit-rate operations (Annex 3)

- At 384kbps, loop filter is essential from picture quality point of view.
- At 1.5Mbps, loop filter is not essential.
- RM6 type (1 2 1) loop filter does not give worse picture quality than others in high bit-rate operations.

(4) Loop filter robustness under IDCT mismatch condition (Annex 4)

- Adaptive loop filters using local characteristics (decoded image pel values) are not robust enough under IDCT mismatch condition.
- RM6 type loop filter is very robust.

3. Consideration

Generally speaking, adaptive filters, such as the filter described in Doc. #356 or the simplified version, give remarkably better picture quality than RM6 type filter. Using local characteristics of a decoded image and other coding results is considered as a key to get good picture quality. In Doc. #207 and #376 from The Netherlands, a filter using decoded image local characterictics was also reported. Unfortunately, the use of decoded image pel values for adaptation causes serious picture quality degradation under the IDCT mismatch condition. This means that such kind of adaptation method can only be used in post processing.
On the other hand, RM6 type loop filter gives sufficient picture quality in high bit-rate operations. In low bit-rate operation, an adaptive post filter can improve picture quality.
As to the block boundary processing, filtering over the boundary is essential if the loop filter is placed before frame memory. In case of the loop filter placed after frame memory, however, exceeding the boundary does not give siginificant improvrment.

3. Conclusion

Extensive study results have been provided for consideration of the meeting. We hope at least the hardware structure is decided at this meeting so that we can start to design the Flexible Hardware.

END.

Annex 1 to Document #406

TITLE: Number and Positions of Filters in Codec
SOURCE: Japan

1. Introduction

This ducument discusses the number and positions of filters used in codec. The following four schemes are tested:

- M1: RM6 type loop filter placed after frame memory (same as RM6)
- M2: an adaptive filter before frame memory
- M3: RM6 + the adaptive filter used as a post filter
- M4: RM6 + the adaptive filter used as a loop filter (before frame memory)

From picture quality and hardware complexity points of view, the followings are summarized.

- M2 fgives a considerably better picture quality than M1.
- M2 gives a slightly better picture quality than M3.
- M2 gives a picture quality equivalent to that of M4.
- the number of filters for good picture quality is minimum with M2.

2. Experiment

Coding simulations have been carried out changing filters. The above four schemes were tested. Figure 1 shows the experimental system. Here, F1, F2 and F3 denote possible filter positions. The following two filters are used in this experiment:

- Type A: same as one in RM6, controlled by MCV.
- Type B: same as described in Doc.#356*, adaptively controlled by block type, local characteristics of image, signficant coefficients and pel position.

In this experiment, improvement by adaptations for RM6 type filter in F1 has not been tested because source informations for adaptive control of filter in F1 are limited.

* an error is included in Doc#356, "(average step size)" should be replaced with "(..)/8.0".

Then the four systems are:

- M1: Type A filter in F1. (same as the original RM6)
- M2: Type B filter in F2.
- M3: Type A filter in F1 and Type B filter in F3.
- M4: Type A filter in F1 and Type B filter in F2.

3. Simulation Results

Table 1 shows average SNYs of the whole sequences. In the table, SNY does not change significantly. Subjective piture quality, however, is improved when the Type B is used. From observations of decoded sequences, the following points are obtained:

(1) Generally speaking, decoded images of M2, M3 and M4 are same in picture quality and better than that of M1. The obvious improvements are found in reduction of block-shape noise and mosquito noise.
(2) The mosquito and block-shaped noises are slightly reduced with M2 and M4 compared with M3.

	Claire	Miss Am.	Salesman	Swing	Blue J.
M1	37.97	37.88	30.79	30.90	33.50
M2	38.20	38.05	30.91	30.64	33.74
M3	38.04	37.94	30.83	30.94	33.55
M4	38.02	37.91	30.82	30.53	33.63

Table 1. SNY (dB)

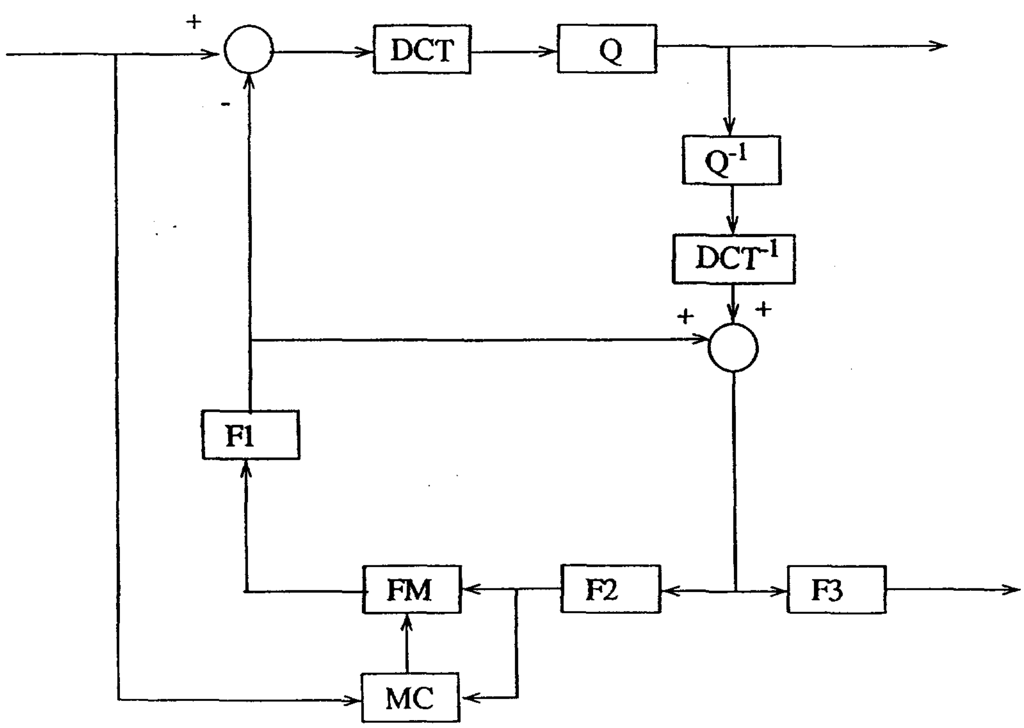


Figure 1. Filter Experimental System

4. Discussions

As a conclusion of Section3, RM6 can be improved in reduction of block-shape and mosquito noise by introdicing an adaptive filter e.g. one described in Doc.#356.
Table 2 shows the number of filters in a set of coder and decoder. Considering the number of filters and performance, M1 and M4 are not attractive solutions. In Table 3, M2 and M3 are compared from performance and complexity points of view. The complexity comparison in the table is based on very rough estimation. The evaluation is rather qualitative and may change accoding to implementations.

5. Conclusion

A set of RM6-based simulations have been carried out in order to study the number and positions of filters in px64kbps codec. From the results, the following points have been concluded:

(1) By using improved loop filter, block-shape noise and mosquito noise observerd in RM6 simulation are reduced. For better picture quality than with RM6, adaptive loop filter or adaptive post filter is required.

(2) From a view point of total complexity to get the better quality, adaptive loop filter is suggested.

END.

	M1	M2	M3	M4
Number of loop filters	2	2	2	4
Number of post filters	0	0	1	0

Table 2. Number of filters

	Picture quality	Number of loop filters	Number of post filters	Filter complexity for compatibility	Filter complexity for good quality
M2	slightly better	2	0	-	+
M3	slightly poorer	2	1	+	-

Table 3. Comparison of M2 and M3

Annex 2 to Document #406

TITLE: Filtering Method on Block Boundaries
SOURCE: Japan

1. Introduction

This ducument discusses the so-called 8x8 and 10x10 issue.

2. Simulation

Simulations have been carried out changing filter characteristics and filtering bounds. Two filters are tested in this experiment. One is the RM6 type filter and the other is a simplified filter(F2M), which is based on the one in Doc.#356. F2M is described in Annex 4 to this ducument.

3. Simulation Result

The simulation result is shown in Table 1. The SNY values with RM6 and those with RM6-10x10 are almost the same. On the other hand, the SNY with F2M-10x10 is 0.2-0.4 dB higher than that with F2M-8x8.

From subjective evaluation, the followings points are summarized:

(1) The deocded sequences with RM6 and RM6-10x10 are very similar. A slight improvement can be observed in RM6-10x10.

(2) The decoded sequence with F2M-10x10 is obviously superior to the one with F2M-8x8. Block-shape and mosquito noises are reduced with F2M-10x10.

4. Hardware Complexity

Estimations of hardware complexity for realizing 8x8 and 10x10 filters depend on implementations. For 8x8 filters, a function to change filter coefficients for filtering of block boundary pels is needed. On the other hand, 10x10 filters require an additional block line memory. In dedicated hardware approach, which might be employed in high bit-rate oper-

	Claire	Miss Am.	Salesman	Swing	Blue J.
RM6	37.97	37.89	30.82	30.87	33.57
RM6-10x10	38.08	37.96	30.79	30.88	33.54
F2M-8x8	37.86	37.77	30.77	30.32	33.48
F2M-10x10	38.25	38.08	31.04	30.50	33.75

Table 1. SNY (dB)

ations, the additional block line memory might be a problem. In a computer-like hardware using DSPs, changing filter coefficients might cost a harder problem than a block line memory because the hardware may have already successive large memories.

5. Conclusion

From the experimental results, the followings can be concluded:

F1 position (1) When RM6 type filter is employed, the picture quality difference between 8x8 and 10x10 is not significant.

F2 position (2) When an improved loop filter, e.g. F2M, is used, the difference is significant and the filter should be 10x10.

(3) Hardware complexity estimation depends on implementations.

END.

Annex 3 to Doc. #406

Hardware Experiments on Loop Filter Characteristics

Using the nx384 kbit/s Flexible Hardware, the following four loop filter characteristics have been compared at 384 kbit/s (p=5 for video rate) and 1536 kbit/s(p=23 for video rate);

Fil-1	Fil-2	Fil-3	Fil-4
1 2.4 1	1 2 1	1 1 1	0 0 0
2.4 2.4 2.4	2 4 2	1 8 1	0 1 0
1 2.4 1	1 2 1	1 1 1	0 0 0

In the Flexible Hardware used, the on/off of loop filter is controlled by the motion vector. Average step sizes and coded picture frame rates were measured as shown in the attached figure for the following four sequences each of which is 25 seconds long;

1) One person close up (Scene B6 in Doc. #363)
2) Split screen with panning
3) Three person seated side by side (Scene B1 in Doc. #363)
4) Three person standing up at the end of meeting (Scene B2 in Doc. #363)

Decoded pictures for the first two sequences are demonstrated in the meeting.

Sequence 1 Fil-4 / Fil-2 / Fil-1 at 384 kbit/s
Fil-4 / Fil-2 / Fil-1 at 1536 kbit/s
Sequence 2 Fil-4 / Fil-2 / Fil-1 at 384 kbit/s
Fil-4 / Fil-2 / Fil-1 at 1536 kbit/s

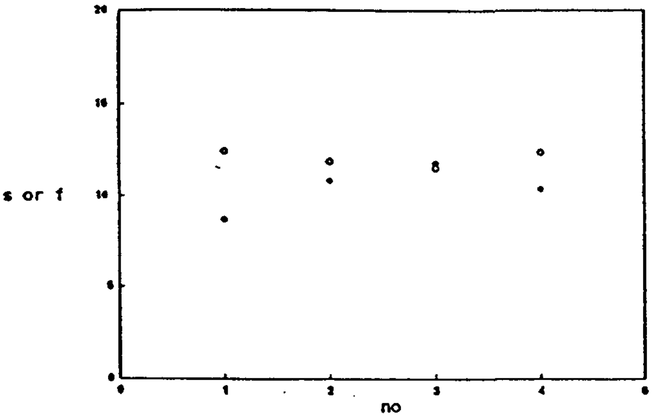
According to the statistical data as well as the observation of the decoded pictures, we can conclude as follows;

- At 384 kbit/s, loop filter is essential to improve coding efficiency. The current 121 filter (Fil-2) or weaker filter gives an optimum. Stronger filters make worse the performance.

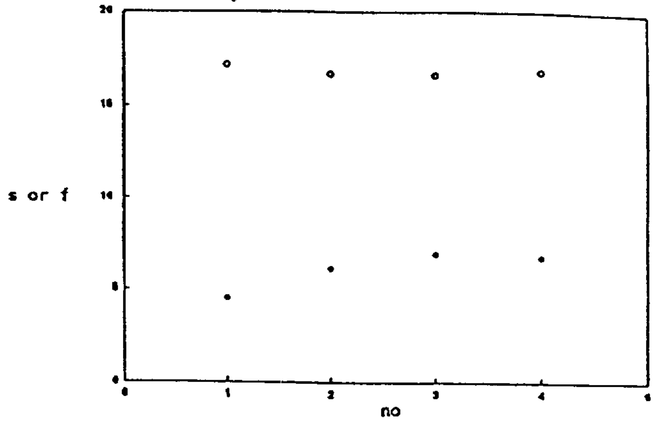
- At 1536 kbit/s (hence at lower step sizes), coding performance is less sensitive to the filter characteristics, including the case without filter and the current 121 case. Stronger filters again make worse the performance.

END

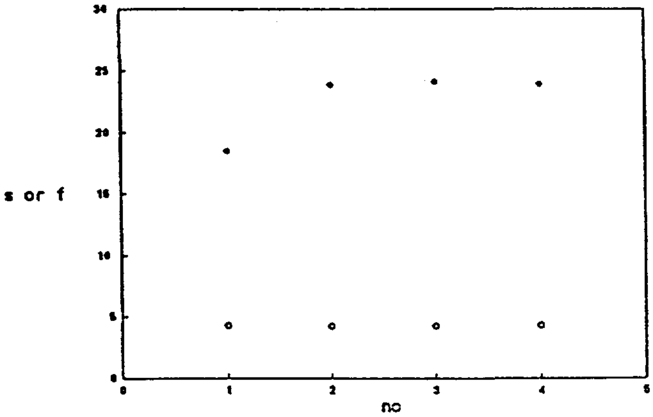
1) 384kb/s



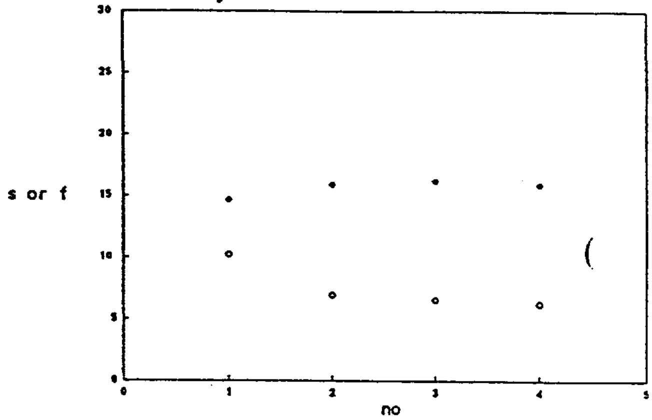
2) 384kb/s



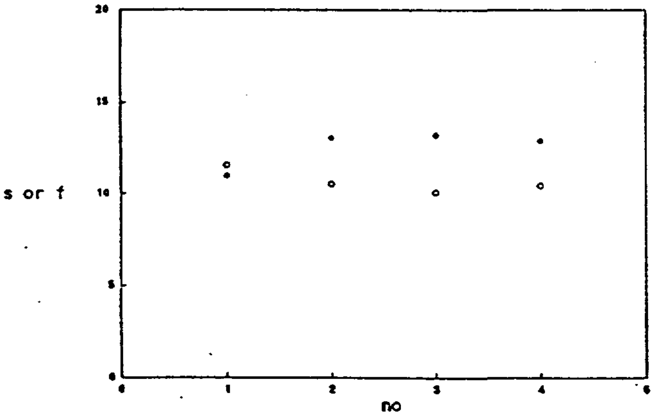
1) 1.5Mb/s



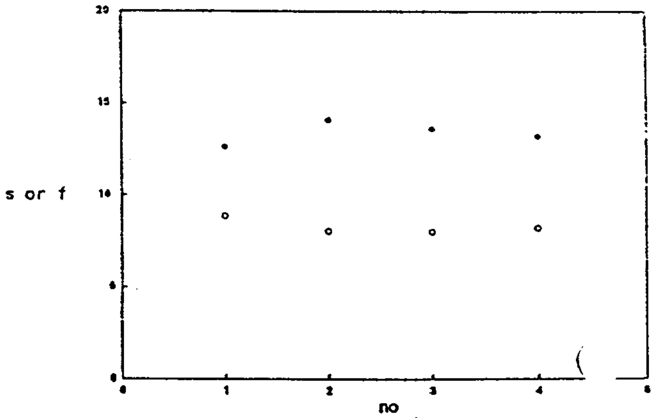
2) 1.5Mb/s



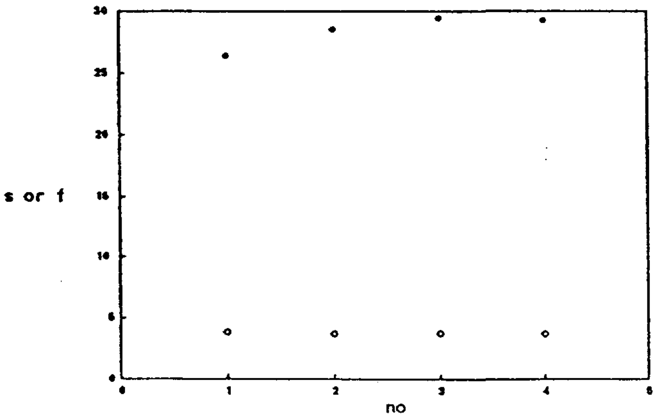
3) 384kb/s



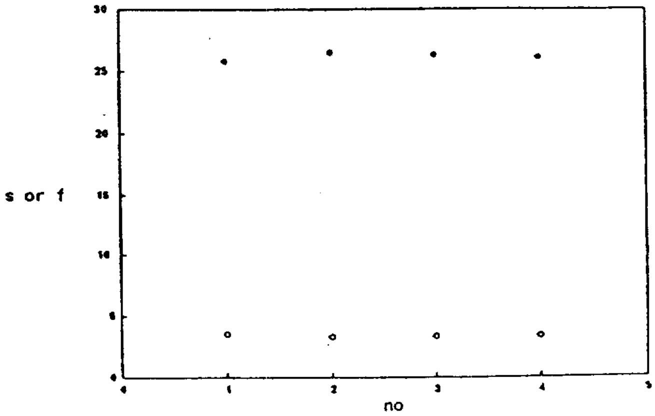
4) 384kb/s_b2



3) 1.5Mb/s



4.) 1.5Mb/s



● frame/s
○ step_size

1	2	3	4
1 2.4 1	1 2 1	1 1 1	0 0 0
2.4 2.4 2.4	2 4 2	1 8 1	0 1 0
1 2.4 1	1 2 1	1 1 1	0 0 0

"ANNEX 4 of Document#406"
Title : Influence of IDCT missmatch
Source : Japan

1. Introduction

In this document influence of the loop filter to decoded images under IDCT mismatch condition is described.

2. Simulation

The scheme to emulate IDCT mismatch is as follows.

MME = $\frac{1}{64}$ =

1) One pixel per one coded block is influenced by mismatch error.
2) The value of mismatch error is +1 or -1 selected by randam number.
3) The location of mismatch pixel is also selected by randam number.

```
if( coded block ) {
      if( (random()%64) < 49) {
            if( (random()%2) == 0) err = 1;
            else                   err = -1;
            block[random()%64] += err;
      }
}
```

Remark: random() is subroutine at 4.3BSD UNIX [0, 2**31-1].
% n represents modulo n.

In this document two types of loop filter are evaluated. One is the same as RM6, and the other is loop filter(F2M) simplified from F2 filter in Doc.#356. For F2 filter was turned out not robust against IDCT mismatch. Simulation are performed at p = 5 for two sequences. The F2M filter algorithm is as follows:

a) *Selection of pixels*

IF{ the block is coded or MCV is not zero } THEN
    the pixels in the block are input to the filter
ENDIF

b) *Filtering process*

All variables are integer.
x(0,0) = Input pixel
{x(i,j),i=-1,1,j=-1,1} Neighbouring pixels
th1 = max[nint{stepsize*1.25/8.0}, 2]
th15 = nint{float(th1)*1.5}
th2 = nint{float(th1)*2.0}
{fil(i,j),i=-1,1,j=-1,1}={1,2,1 2,4,2 1,2,1}

$$yo = [\{ \sum_{i,j} x(i,j)*fil(i,j) \} + 8] / 16$$

```
y(1) = { x(-1,-1) + x(0,0) * 2 + x( 1, 1) + 1 } / 4
y(2) = { x(-1, 1) + x(0,0) * 2 + x( 1,-1) + 1 } / 4
y(3) = { x( 0,-1) + x(0,0) * 2 + x( 0, 1) + 1 } / 4
y(4) = { x(-1, 0) + x(0,0) * 2 + x( 1, 0) + 1 } / 4
z(k) = absolute( y(k) - x(0,0) )
ym = y(k), where z(k) = min z(l)
                        l = 1,2,3,4

IF{ absolute( yo - x(0,0) ) less than th1 } THEN
    output = yo
    ELSE IF{ absolute( ym - x(0,0) ) less than th2 } THEN
        output = ym
    ELSE
        IF{ ym > x(0,0) } output = x(0,0) + th15
        ELSE output = x(0,0) - th15
ENDIF
```

```
121            100   001   010   000
242            020   020   020   121
121            001   100   010   000
 yo            y(1)  y(2)  y(3)  y(4)
```

Filter coefficients for calculating yo, low pass filter.

Filter coefficients for calculating y(i), edge preserving filter.

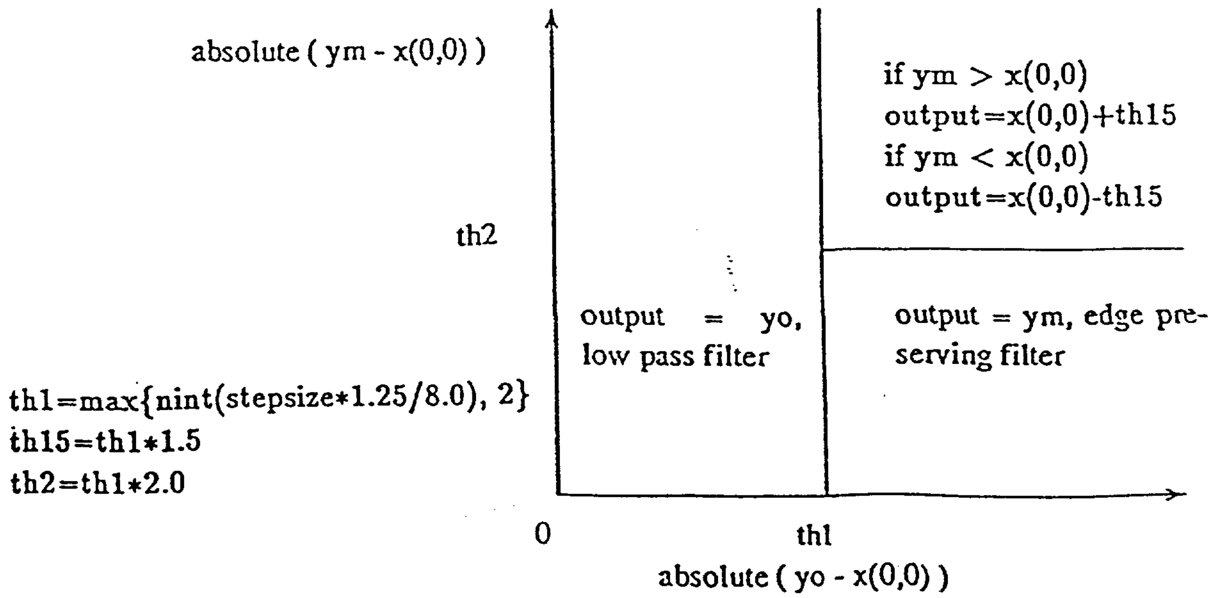


Figure 1: F2M filter adaptation by stepsize

3. Conclusion

The simulation results are shown in the table 1,2 and figure 2,3. From the results F2M filter is weak in respect of the immunity against IDCT mismatch. It seems that adaptive processing controled by decoded pixel value is not sufficiently robust against IDCT mismatch.

TABLE 1

STATISTICS RM6 DATE : 1988. 11. 29
SEQUENCE :claire(1-164)
MODIFICATION :IDCT mismatch FRAMERATE : 10Hz

Modification (mean)		RM6	RM6 miss	F2M	F2M miss
RMS for Y		1. 61	1. 62	1. 55	1. 67
SNR for Y		43. 97	43. 97	44. 34	43. 67
SNR for U		43. 82	43. 81	43. 92	42. 15
SNR for V		46. 37	46. 36	46. 51	45. 85
SNR for C		44. 91	44. 90	45. 02	43. 62
MEAN STEP SIZE		5. 60		5. 35	
MEAN Numb. of NONZERO Coeff.	Y	5. 87		5. 76	
	C	2. 84		2. 79	
MEAN Numb. of ZERO Coeff.	Y	9. 83		9. 48	
	C	7. 50		7. 34	
Block Type of MACRO	INTRA	0		0	
	FIXED	143		145	
	CODED MC	48		47	
	FIXED MC	0		0	
	CODED	205		203	
Block Type of Y	FIXED	1031		1039	
	CODED MC	174		170	
	FIXED MC	20		20	
	CODED	360		354	
Block Type of UV	FIXED	626		618	
	CODED	166		174	
MACRO ATTRIB.		1743		1745	
End of Block		1953		1943	
Motion Vector		258		254	
INTRA	DC	1		1	
Coeff.	Y	19359		18586	
Coeff.	UV	2618		2671	
Coeff.	TOTAL	25932		25200	

γ 0.68

TABLE 2

STATISTICS RM6 DATE : 1988. 11. 29
SEQUENCE :Salesman(1-149)
MODIFICATION : IDCT mismatch FRAME RATE : 10Hz

Modification (mean)		RM6	RM6 miss	F2M	F2M miss
RMS for Y		3. 93	3. 93	4. 44	5. 33
SNR for Y		36. 24	36. 24	35. 43	33. 59
SNR for U		41. 79	41. 79	41. 65	41. 07
SNR for V		42. 81	42. 81	42. 61	42. 12
SNR for C		42. 27	42. 27	42. 11	41. 57
MEAN STEP SIZE		10. 13		9. 84	
MEAN Numb. of NONZERO Coeff.	Y	4. 52		4. 47	
	C	1. 70		1. 59	
MEAN Numb. of ZERO Coeff.	Y	26. 05		22. 06	
	C	3. 07		2. 87	
Block Type of MACRO	INTRA	2		2	
	FIXED	126		102	
	CODED MC	61		51	
	FIXED MC	0		1	
	CODED	207		239	
Block Type of Y	FIXED	856		774	
	CODED MC	216		172	
	FIXED MC	30		37	
	CODED	481		601	
Block Type of UV	FIXED	720		711	
	CODED	72		81	
MACRO ATTRIB.		1658		1823	
End of Block		1916		2134	
Motion Vector		342		295	
INTRA	DC	14		16	
Coeff.	Y	25148		24773	
Coeff.	UV	609		618	
Coeff.	TOTAL	29687		29659	

γ 1.34

Fig.2 : RM6 vs F2M vs F2M.MISMATCH

(p=5, 10Hz, 164frames, Claire)

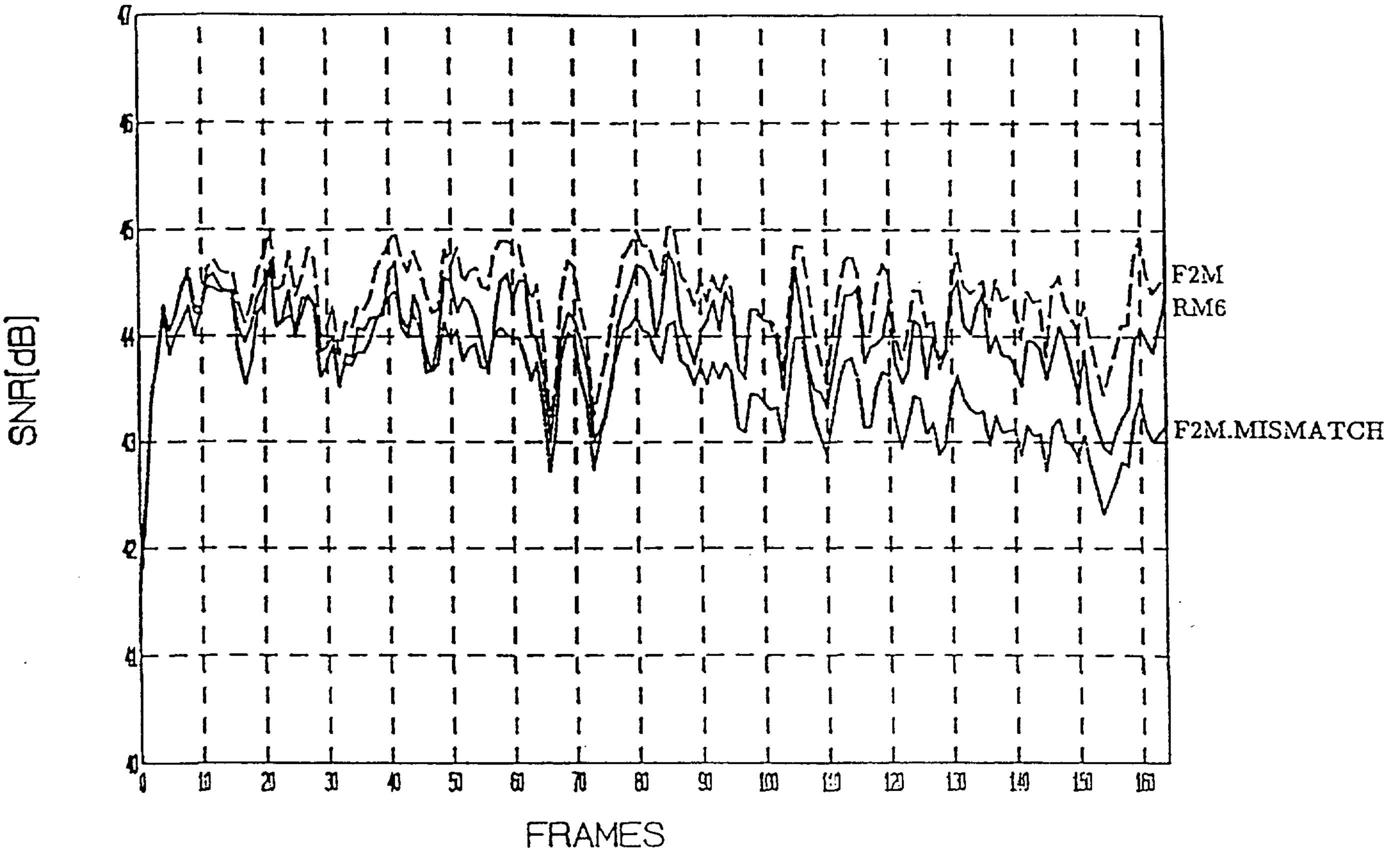
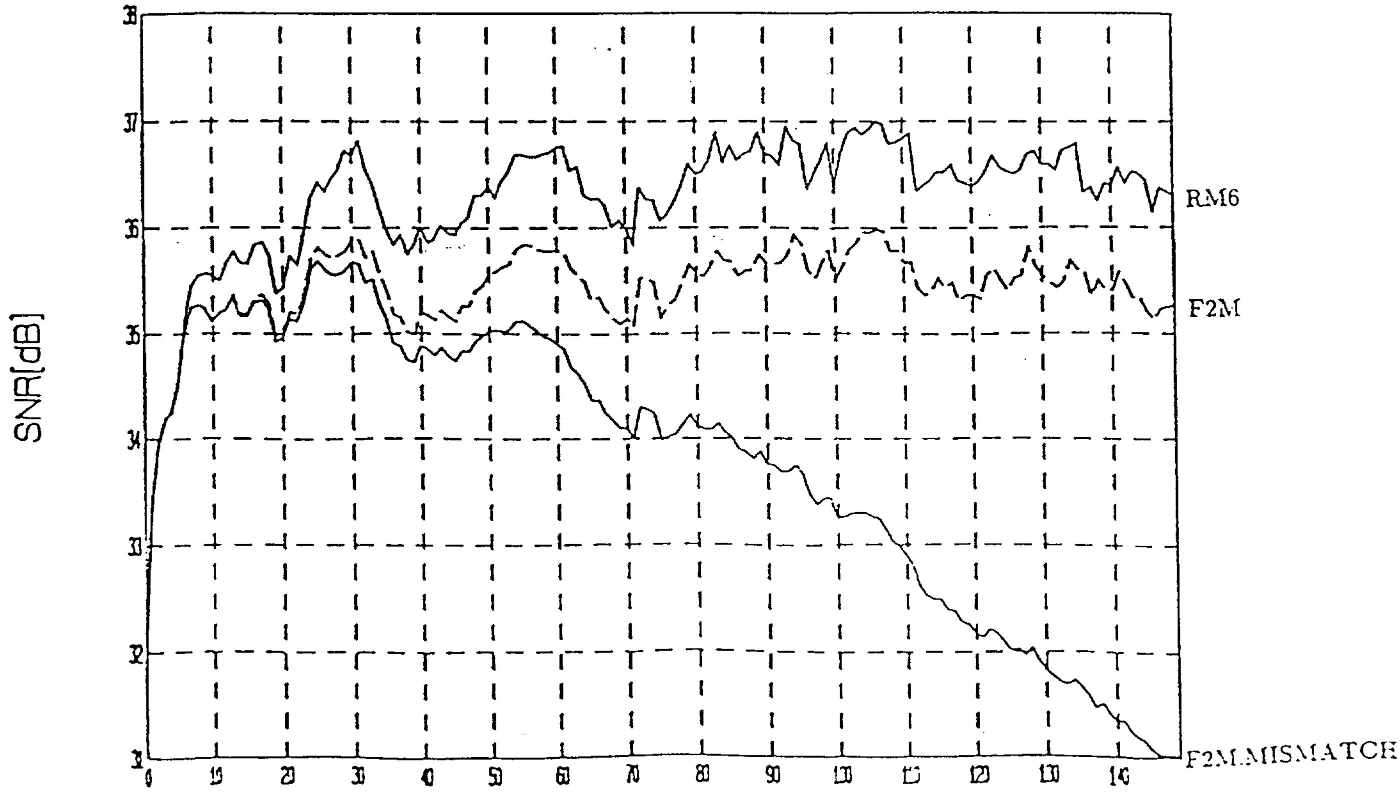


Fig.3 : RM6 vs F2M vs F2M.MISMATCH

(p=5, 10Hz, 149frames, Salesman)



FRAMES

TABLE 3-1

	Claire								Miss America							
	2 nd		3 rd		15 th		Mean		2 nd		3 rd		15 th		Mean	
	RM 6	Impro	RM 6	Impro	RM 6	Impro	RM 6	Impro	RM 6	Impro	RM 6	Impro	RM 6	Impro	RM 6	Impro
1. RMS for Luminance	9.38	6.79	4.89	4.92	3.06	3.04	3.19	38.14	4.83	4.79	4.30	4.19	3.04	3.01	3.25	3.20
2. SNR for Luminance	28.68	31.48	34.33	34.27	38.41	38.45	38.04	38.18	34.43	34.50	35.45	35.67	38.44	38.54	37.86	38.02
for Chrominance (U)	34.90	35.18	35.86	35.86	39.13	39.15	38.96	39.96	36.36	36.39	36.84	36.86	38.30	38.40	37.86	37.97
for Chrominance (V)	38.71	38.69	39.04	39.12	42.07	42.00	42.18	42.16	35.63	35.72	36.99	37.12	39.67	39.67	38.76	38.99
3. Mean value of step size	56.55	56.44	48.88	48.55	16.00	15.66	18.79	18.61	54.00	53.22	39.44	37.77	16.33	15.77	19.70	18.72
4. Mean value of the number of non-zero coefficients Y	2.63	2.68	2.13	1.93	2.88	2.75	3.14	3.16	1.91	1.91	1.84	1.64	2.53	2.77	2.32	2.38
C	1.23	1.23	1.07	1.09	1.61	1.37	1.66	1.67	2.04	2.09	1.60	1.67	1.39	1.25	1.38	1.43
5. Mean value of the number of zeros before the last NZ-coefficient Y	3.20	3.24	7.13	8.37	6.43	5.89	5.86	5.95	1.59	1.60	2.93	2.48	4.06	3.84	3.71	3.86
C	0.80	0.80	1.42	1.29	4.06	4.17	2.77	2.88	3.66	3.62	2.88	3.11	1.64	1.40	1.95	2.15
6. Block type of macro: Fixed	286	281	270	266	262	248	276	275	254	252	270	264	231	222	212	218
Coded MC	80	82	44	40	29	20	47	38	81	81	33	36	44	31	76	48
Fixed MC	19	22	24	22	6	1	6	5	43	43	12	13	9	4	22	12
Coded	9	9	57	67	99	127	68	78	18	20	80	82	112	139	86	118
Intra	2	2	1	1	0	0	0	0	0	0	1	1	0	0	0	0
7. Block type of Y: Fixed	1166	1146	1225	1241	1278	1241	1267	1277	1065	1062	1288	1260	1273	1314	1114	1209
Coded MC	223	230	81	71	83	53	125	99	149	153	41	53	99	64	132	81
Fixed MC	173	186	191	177	57	31	86	72	347	343	139	143	113	76	261	158
Coded	22	22	87	95	166	218	106	136	23	26	116	128	99	130	78	135
8. Block type of UV: Fixed	771	771	754	751	761	752	755	753	750	749	721	724	688	682	683	671
Coded	21	21	38	41	31	40	37	39	42	43	71	68	104	130	109	121
9. Number of bit: Macro attributes	601	703	798	789	921	851	793	706	782	879	908	820	1180	1104	1212	1112
End of block	680	696	632	632	804	896	762	778	648	668	752	812	968	1072	1012	1064
Motion vectors	1142	832	541	550	187	99	310	261	1421	992	364	409	277	204	606	387
DC coefficients	16	16	8	8	0	0	1	1	0	0	8	8	0	0	1	1
Coefficients Y	3355	3535	2028	1945	3728	3818	3767	3872	1465	1527	1303	1344	2645	2766	2370	2526
Coefficients U	98	98	134	135	166	210	191	204	86	91	181	172	375	433	296	357
Coefficients V	24	24	43	55	80	85	86	88	349	360	398	416	208	224	342	391
Coefficients TOTAL	3477	3657	2205	2135	3974	4113	4044	4165	1900	1978	1882	1932	3228	3423	3008	3274
TOTAL	5916	5904	4184	4114	5886	5959	5910	5910	4751	4517	3914	3981	5653	5803	5838	5838