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

Document #114
June 1986

Source: NTT, KDD, NEC and FUJITSU

TITLE : CODING OF MC VECTORS

1. Introduction

In this contribution, two motion vector encoding methods are compared with respect to error resilience, coding efficiency and hardware complexity. Along with this comparison, a proposal for selection of one out of two motion compensation schema with integral or fractional vectors is described. As a result, it is proposed that motion vectors with integer accuracy are encoded in the form of difference between two successive vectors.

2. Accuracy of Motion Vector Detection

If motion vectors with fractional accuracy are to be used, the number of vectors/unit area will increase by a factor of four. In this motion compensation scheme, interpolation is necessary to produce sample values between integral sample points. For half-integer accuracy detection, the number of data to be used in the detection is four times larger. To produce these interpolated values, hardware vector detector should have four times faster operation speed than the original sampling speed (6.75 MHz). This makes the hardware much more complex than that without interpolation. The number of calculations needed only for vector detection may be the same by retaining the total number of trial vectors actually used.

What is most important here is the operation speed. If parallel computing scheme is adopted to overcome this problem, the complexity of vector detection hardware will be much more than doubled, maybe 3 --4 times larger.

Therefore, motion compensation with half-integer accuracy is not preferable. Coding efficiency improvement is not discussed here. But such hardware complexity increase would be paid off only by, for example, two-fold efficiency improvement.

3. Two Motion Vector Encoding Methods

(1) Definition

PCM Method :	Motion vector is encoded with transform coefficients on a block-by-block basis as a part of Block Attribute in the form of absolute (PCM) value as described in Doc. #103R Annex 3.
Difference Method :	Motion vector is encoded collectively as

a part of GOB data in difference form between two successive vectors. Zero-difference vectors are encoded by run-length coding.

(2) Comparison

The two methods are compared with respect to error resilience, efficiency and hardware complexity.

Huffman coding is assumed here not only for non-zero vectors but also for non-zero difference vectors. Successive zero-difference vectors are runlength-coded in the Difference Method.

1) Error Resilience

If channel error occurs in the bit stream corresponding to vector information, result will be different depending upon coding schemes.

PCM Method :

a) Temporarilly Decodable Case

If a bit error occurs in a limited part of Huffman codes for vectors, decoding may be completed , though an incorrect vector is reproduced (i.e. seemingly consistent anyway). The resulting picture shows localized block error running around after the moving object. Affected by the motion direction, error blocks increase or decrease in number as time passes by. In general, this block error does not disappear.

This happens only when there exists a Huffman code corresponding to the temporarily-decoded Huffman code.

Example of Temporarily-decodable Codes

```
Code A = 1 0 1 1 1 0
                 ↓ Bit Error
Code B = 1 0 1 1 0 0
```

When the 5-th bit of Code A is changed to "0" due to channel error, this code is decoded as Code B. Decoding can be carried out seemingly with consistence.

Discussion on Decodability of an Erroneous Huffman Code

An example of a variable word-length code set for motion vectors is attached in the last page of this document. Similar code set can be designed for PCM as well as for Difference vectors. Therefore, the analysis in what follows holds for the two coding methods.

According to the table, decodability remains with a probability ranging from 3/5 to 1/4, provided that a single error occurs uniformly in vector information.

Example of Approximated Probability for Decodability

Code-length = 5		Probability
1 1	1 Sx Sy	3/5
1 1	0 1 Sy	
1 1	0 0 Sx	
	Decodable even if any one bit after this line is changed.	

If motion vector information occupies 1/2 to 1/4 of the total information amount to be transmitted, the erroneous Huffman codes would be seemingly decodable only with a low probability ranging from 3/10 to 1/16 (3/5 x 1/2 to 1/4 x 1/4).

This seemingly consistent "Decodability" requires special care in designing such Huffman code set to have as many similar code groups as possible where a single bit error does not affect the " decodability".

b) Undecodable Case

Except the case described above, decodability is not assured.

Difference Method :

Vector information after a bit error has occurred can not be correctly decoded. Any error in differential vector will result in off-set to the following decoded vectors even if the error does not affect the decodability.

2) Efficiency

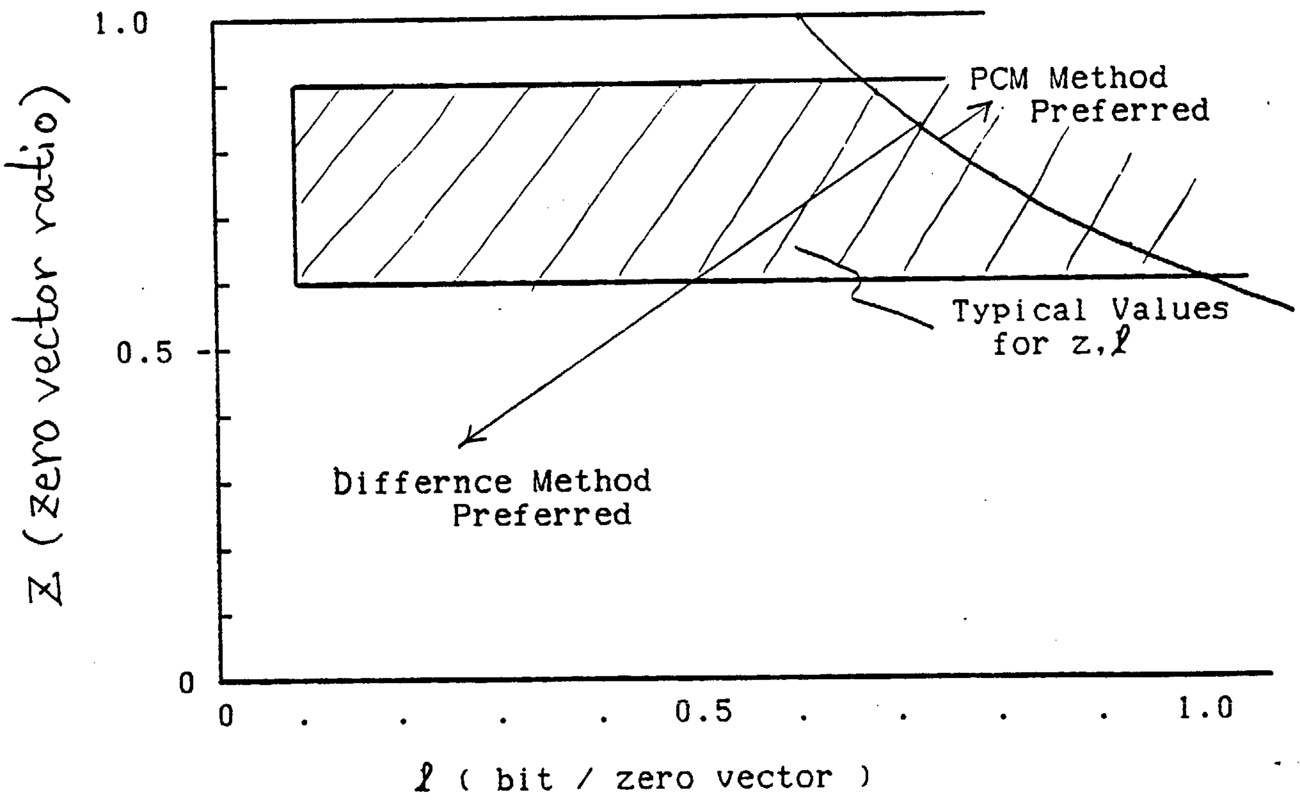
a) Encoding of Zero-vector

In encoding videoconference sequences, zero-vector information occupies about 50 to 90 % of the total motion vector information. This property should be fully used to reduce the motion vector information.

In PCM Method zero-vector information is encoded as a part of Block Attribute(e.g. Reference Model), while it is encoded as a part of GOB data in the Difference Method(e.g. Part-3/H.120).

Information Amount :

	PCM Method	Diff. Method
Block Type	2 bit/Y-block 1.4 bit/C-block $\longrightarrow$237.6 bit/GOB	1.4 bit/block $\longrightarrow$184.8 bit/GOB
Zero-vector Data z:Zero-vector l:code-length	—	zl bit/block $\longrightarrow$185+88zl bit/block
Total	238 bit/GOB	185+88zl bit/GOB



Assuming that typical values of z and ℓ are 0.6 and 0.5, respectively, bit saving achieved by the Difference Method is about 27 bit/GOB i.e. 486 bit/frame. (Code-length table for zero-vectors adopted in Part-3 codec/H.120 is also exemplified in the last page of this document.)

According to this calculation example, there is no remarkable difference in efficiency though the Difference Method gives a little bit smaller information rate.

b) Encoding of Vectors in Panning Sequences

Even if motion vectors are detected for panning pictures, usually they are not represented by a single vector. Particularly, many different correct or incorrect vectors are detected in the vicinity of contour lines.

No effective method for detecting the single representative vector (Global Vector) is obtained. Anyway, Global Vector detection requires a lot of computations including incorrect vector discrimination.

When a local vector actually detected for each block is slightly different from Global Vector, the slight vector difference usually results in coding efficiency degradation.

Panned pictures may include large-area motions with vectors opposite to panning direction.

c) Encoding of Vectors for Large Area Motions

There are many incorrect vectors in the vicinity of contour lines as in panning example. However, similar or same vectors are usually found inside the contour lines. Many bits are necessary to represent these non-zero vectors inside the contour lines. The same is true of motions which seem translational. Encoding the difference vectors will help reduce vector information amount.

(3) Hardware Complexity

As long as non-zero vectors are encoded with Huffman codes, hardware complexity is hardly increased by introducing run-length coding of zero-difference vectors since run-length detection does not matter and Huffman codes for run-lengths can be easily designed and implemented. Decoder hardware for vector information is also simple since decoding operation is 1/64 times slower than for encoded transform coefficients.

4. Conclusion

(1) Vector Detection Accuracy

Since hardware complexity increases due to very fast operation (equivalently 4 x 6.75 MHz) needed in motion vector detection, motion compensation with fractional vectors is not recommended.

(2) Motion Vector Encoding Method

Essentially, error resilience is not assured in motion-compensated interframe coding. With respect to coding efficiency, the Difference Method is preferred since a single vector cannot represent an object motion even if the motion is seemingly translational, and wrong vectors degrades coding efficiency to a large extent. Moreover, Global Vector detection method is not well investigated.

Run-length coding is suitable for efficient coding of successive zero vectors. It does not increase hardware complexity of VLC codec for the PCM Method as long as Huffman coding is used for non-zero vectors both the PCM and Difference Methods. Run-length codec can be implemented in the same Huffman codec.

Therefore, it is concluded :

a) To encode difference vectors between two successive blocks.
b) To encode vector information as a part of GOB data with zero-difference vectors runlength-coded.

Concerning Global Vector information, it may be encoded as a part of Picture and/or GOB attributes for future use.

TABLE[illegible]. Variable length code and run length code for motion vector data

ΔVx	ΔVy	Code Length	Code Word		Number of Code
±1	0	4	001Sx	1/4 ビット bit	2
±1	±1	5	111SxSy		
0	±1	5	1101Sy	3/5 ビット	8
±2	0	5	1100Sx		
0	±2	6	10111Sy		4
±3	0	6	10110Sx	2/6 ビット	
±1	±2	7	10011SxSy		
±2	±1	7	10010SxSy	4/7 ビット	12
0	±3	7	100011Sy		
±4	0	7	100010Sx		
±3	±1	9	1010111SxSy		
±1	±3	9	1010110SxSy		
±2	±2	9	1010101SxSy		
±3	±2	9	1010100SxSy		
±4	±1	9	1010011SxSy		30
±4	±2	9	1010010SxSy	5/9 ビット	
±5	0	9	10100011Sx		
±6	0	9	10100010Sx		
0	±4	9	10100001Sx		
-8～7	-5～+5 (See Fig.13)	11	100001XXXX Sy	[X]=ΔVx 5/11 ビット	32
-16～15	-6～+6 (See Fig.13)	13	0100001XXXXX Sy	[X]=ΔVx 6/13 ビット	64
-16～15	-8～+7 (See Fig.13)	15	100000XXXXXYYYY	[X]=ΔVx [Y]=ΔVy	360

9/15 ビット

RL	Code Length	Code Word		Number of Code
1	3	000		1
2	4	0111		1
3～6	6	0110XX	XX= 6-RL	4
7～12	7	0101XXX	XXX=12-RL	6
13～20	8	01001XXX	XXX=20-RL	8
21～28	9	010001XX	XXX=28-RL	8
TRANS	6	010111		1

Note: Sx and Sy denote signs. Si=0 for positive, Si=1 for negativ