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

Document #113
June 1986
English Only

Source : NTT, KDD, NEC and FUJITSU

Title : VIDEO MULTIPLEX

1. Introduction

In this contribution, we shall discuss data structure and error resilience to channel error from the view point of hardware complexity and decodability of received data as well as error localization.

As a result, collective data handling with forward error correction in conjunction with forced update is recommended.

2. Data Structure in GOB

Since picture attribute and fixed information in GOB (e.g. GBSC and GN) hardly affect hardware complexity or coding efficiency, this section is devoted to discuss the structure of data which is essentially variable and encoded within GOB,

(1) Information to Be Encoded within GOB

MCV : Motion Vector
BA : Relative Block Address
BDATA : Block Type (f, p, v, t), Classification Index
QDATA : Quantized Coefficients

(2) Order of Encoding/Transmission

Scheme 1) : MCV, BA, BDATA and QDATA are encoded on a block-by-block basis as block data. No correlation is used in these individual components.

Scheme 2) : MCV, BA and BDATA are encoded individually in collective forms as a part of GOB data. Correlation can be expected and employed in encoding each information. For example, motion vector (MCV) is encoded in a difference form to reduce information amount. Coefficient information (QDATA) is encoded as Block data.

(3) COMPARISON

Scheme 1) : In real time processing, encoding of these data should be finished within one block time (N x N samples). In many cases, 3 -- 6 bytes data should be processed within one clock time (

1/6.75 micro sec), since the total Huffman code amount of MCV, BA and BDATA in a block will be encoded at a predetermined time slot (clock) along with coefficients (QDATA), simultaneously.

Example

```
MCV   :  4 -- 15 bits
BA    :  3 -- 12 bits
BDATA :  9 -- 11 bits
           [ 2 bits to (f, p, v, t),
            4  -- 6 bits to Classification Index]
QDATA :  1 -- 10 bits for each DC/AC coefficient
-------------------------------------------------
Total : 14  -- 45 bits  (3 -- 6 bytes)
```

- For this assignment example, it is probable that information amount of 3 to 6 bytes is generated at the beginning of a block. Bit rearrangement from 6 bytes to 1 byte in a clock time in VLC followed by Buffer Memory requires very complex hardware. However, if maximum data generation rate is limited to be no more than 1 byte /clock time or 8 x 64 (Block Size) bit/block, the hardware complexity will be much reduced since a small buffer memory will suffice for the rate smoothing.

- Decoding control at the decoder is complicated since five kinds of data are mixed up in a small period of time i.e. block. Very fast decoding speed will be needed.

- Difficulty will arise in designing Huffman codes for these data. (Pseudo-sync code should not appear for any possible combination of these Huffman codes. --- Efficiency degradation ?)

- No buffer memory with a GOB capacity is needed.

Scheme 2) More efficient VLC coding for collective data employing correlation in each data.

- Easy Huffman code design (Pseudo-sync code can be avoided much more easily.)

- In effect, no processing time limitation is imposed, though five buffer memories each with a GOB capacity are needed at most.

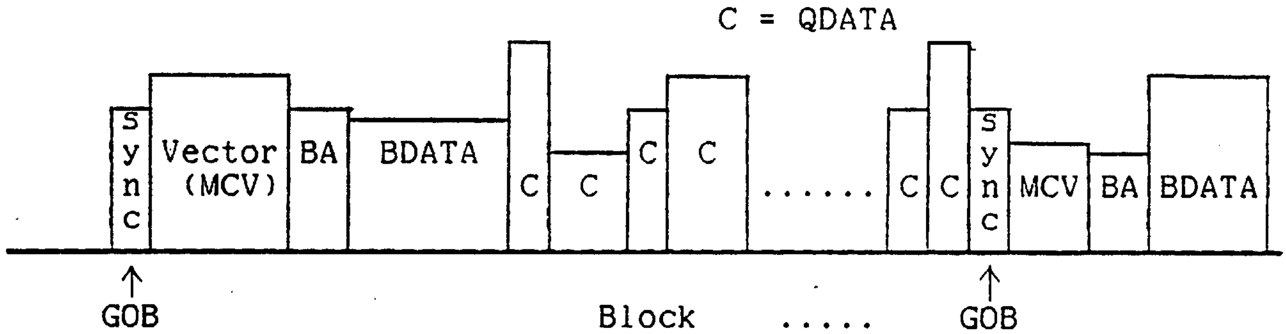
- Simpler VLC decoding cotrol.

Result

There is no paramount difference in coding efficiency between the two schemes. However, "rearranging 3 -- 6 bytes data into 1 byte data within a clock time " requires considerable hardware. This will be the largest difficulty in adopting the Scheme 1).

On the other hand, five buffer memories at most will be easily implemented using state-of-the-art LSI FIFO memories.

In conclusion, Scheme 2) where block attribute data is encoded in collective forms as a part of GOB data is recommended. This scheme is exemplified in the following figure.



3. Error Resilience

Assuming that video data is encoded and transmitted in the order shown by the figure above, this section discusses channel error effects on each component of transmitted data. Decodability is also discussed, and error localization as well.

Decodability falls into two categories; one is a category where correct decoding is carried out, while the other is a provisionally-decodable case where a Huffman code exists, though possibly meaningless, corresponding to a Huffman code with a bit error. Otherwise, decoding is never carried out further usually resulting in discarding the remaining data until the next Picture Start Code appears.

1) Channel Error in MCV

This is discussed in detail in Document #114 CODING OF MC VECTORS. Briefly, the result is that block error propagates in the reproduced sequences even if there exists a motion vector corresponding to a Huffman code with channel error. In general, true decodability can not be expected in case of channel error as long as Huffman coding is employed.

2) Channel Error in BA

After provisional decoding of significant block position, inverse-transformed coefficients are added to samples in blocks at wrong positions at the receiver. Due to motion compensation, this block error will be running about after moving objects corresponding to motion vector and in general will never disappear. That is, error propagates in two dimensions. Channel error in BA results in block error in relatively broad area.

3) Channel Error in BDATA

(a) Error in Block Type (f, p, v, t) : The reproduced sequences are similar to the result in case of channel error in BA, if BDATA with bit error is provisinally-decodable. However, it should be noted that due to channel error there might be inconsistent cases of Intraframe with motion compensation, no coefficients information for significant blocks, (non-MC) Interframe prediction with motion vectors, etc.

All these results lead to block error running around in the reproduced sequences.

(b) Error in Classification Index : The result depends upon what this index represents. If it indicates quantizer selection, wrong coefficient values are obtained provided that they are provisionally-decodable. This leads to block error possibly confined in a block. If it represents zone pattern including scanning method in VLC coding, block error will propagate over many blocks. It will be running about in motion compensation.

(c) Error in Quantizer Type : Wrong coefficient values are reproduced resulting in a block error confined in a block. The block error will be running about corresponding to motion vectors and usually propagate in two dimensions.

4) Channel Error in QDATA

Due to a single error, a single coefficient will be in error in the provisionally-decodable case. Seemingly, reproduced sequences may not show any error effect due to the property of transform coding i.e. error effect is diffused over a block. However, mismatch between coder and decoder exists even if it is invisible, block error becomes visible sooner or later.

(Conclusion)

Most important issue in Error Resilience will be the " provisional decodability " and error localization discussed above.

Without motion compensation, error in reproduced sequences might be suppressed by " leaky " integration in the temporal axis. Error might disappear in a predetermined time specified by the leaky factor. With motion compensation, the error moves to a different position before it disappears. To avoid this, the leaky factor should be much smaller than unity. However, this will inevitably result in bad prediction.

Therefore, error correction should be considered for use in the Subrate codec. Assuming that (4095,4035)BCH code is employed, random error rate of one in 10 is improved to that of 6.5 x 10 (mean error free time of 113 hours at 384 kbit/sec).

4. Number of VLC Code Sets

The number of VLC code sets essentially has nothing to do with error resilience. As long as the " provisional decodability" is assured with some probability, the problem is whether or not the probability is high. The number of VLC code sets is not the problem. Even in a single code set, error resilience can not be expected if the Huffman code set is not well designed so as to give high provisional-decodability. Since this kind of Huffman code set will have high redundancy, coding efficiency degradation will be of problem.

Essentially, plural code sets can be employed without increasing vulnerability to channel error. Error correction should be applied for resilience to channel error.

5. Forced Update

According to the discussion described above, error resilience can not be expected as long as Huffman coding is employed. This leads to employment of error correction coding. Furthermore, error correction is not sufficient for transmission through commercial links, regardless of terrestrial or satellite links. Error line substitution will not suffice, either. To cope with this difficulty, video data should be refreshed periodically and/or on demand.

(1) Forced Update

Forced Update is necessary in broadcast type transmission and is carried out throughout a GOB or plural GOB's using Intraframe operation. Update speed depends upon transmission bit rate.
At low rates for example 384 kbit/sec, one GOB/frame will be to much if it is used periodically. Assuming that 10 % of video data is assigned to Forced Update and Intraframe operation with 0.5 to 1 bit/pel is achieved, refresh time is approximated to be 2.5 to 5 sec.

(2) On-Demand Refresh

When backward channel is available, on-demand refresh request is signalled in codec-to-codec information from receiver to transmitter. It is used to build up a new picture at receiver in a short time.
On-demand refresh capability can also be used for fast build-up purpose in switched multi-point connection possibly under the control of Multipoint Conference Unit.

Both refresh capability is realized by Intraframe operation which is a part of coding algorithm currently under study. This means hardware complexity does not increase.

When refresh operation is evoked, it can be signalled as a part of Picture or GOB attribute information by the encoder.