CCITT SGXV
Working Party on Visual Telephony
Specialists Group on Coding for Visual Telephony

Doc. # 102

March 86

Source: UK, Sweden and France

Title: Video multiplex for nx384kBit/s

1. Introduction

In the design of any video multiplex there is a compiromise between absolute efficiency and error performance. Most simuilations to date have been concerned only with efficiency. An example of this is to simply allocate one bit per block to indicate whether any particular block has data associated with it. In the case of a transmission errors whole fields of data are lost which is catastrophic in a codec which relies heavily on interframe coding. The opposite extreme is to give each block an absolute address within each field. In this case efficiency suffers.

The following proposal is a fairly simple compromise where absolute addressing is used for each line of blocks and a more efficient relative addressing mode is used between blocks. This will contain errors to within one or two lines of blocks without significantly reducing efficiency.

2. Possible Video Multiplex Arrangement

i. Picture Start Code (PSC)

1000 0000 0000 0000 0000 [Buffer State] [Temporal Ref.] [Type]

Buffer State :- 6 bit number representing the encoder buffer fullness in 1kBit intervals at the beginning of the current picture.

Temporal Ref. :- A three bit number representing the time sequence in 1/30 sec. intervals of a particular picture. All Picture Start Codes should be transmitted.

Type :- This is a VLC code which allows block attributes to be applied to all blocks within a field (eg. all blocks may be intraframe coded or non-motion compensated). This saves overhead by removing the requirement to signal Block Type on a per block basis. Possible attributes are shown in appendix 1.

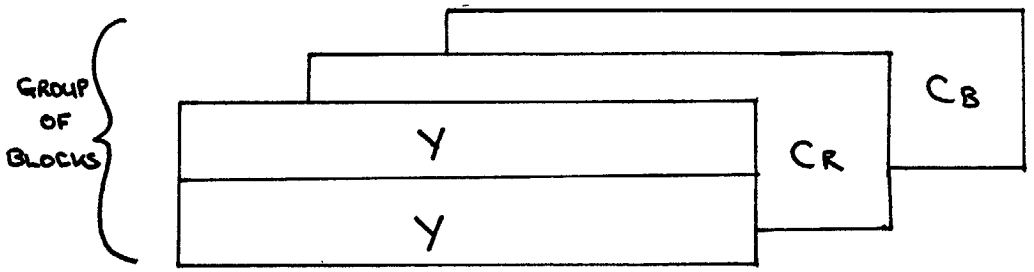
NB. The exact bit pattern and the word length of the Picture Start Code will depend on the VLC code set chosen and is for further study.

ii. Group of Blocks Start Code (GBSC)

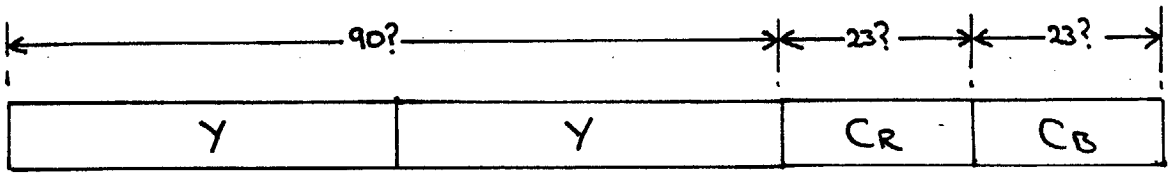
1000 0000 0000 1111 [GBSC Number] [Type]

Block Line Number:- A six bit number representing the vertical spatial position in blocks of the current group of blocks. All line of block codes should be transmitted.

NB: A group of blocks consists of two lines of luminance blocks, one line of U blocks and one line of V blocks. This will mean that a Group of Block Start Code defines the beginning of a physical region of picture (luminance and colour differences) outside of which errors are unlikely to extend. (See below)



PICTURE INFORMATION CONTAINED IN A GROUP OF BLOCKS



ORDER OF BLOCKS TRANSMISSION

NB: CR AND CB ARE COLOUR DIFFERENCES

The exact bit pattern and the word length of the Group of Blocks Start Code will depend on the VLC code set chosen and is for further study.

Type :- This is a VLC code which allows block attributes to be applied to all blocks within a group of blocks (eg. all blocks may be intraframe coded or non-motion compensated). This saves overhead by rtemoving the requirement to signal Block Type on a per block basis. Possible attributes are shown in appendix 1.

iii. Block Address

[Block Address]

A VLC code indicating the relative piosition of a moving block (relative to the previous block or picture boundary).

Addresses of absolute position greater than 90 are considered to be chrominance blocks.

iv. Block Type

[block type]

A VLC code representing the type of the block (possible attributes are described in doc.69 France). Possible types are:-

i. Intraframe coded block
ii. Interframe coded block
iii. Motion compensated block
iv. Motion compensated with coded residue
v. Future expansion on
vi. Future expansion off

Codecs should be designed to ignore all data between types v. and vi., also between type v. and the next LBSC. This will allow us to efficiently include some enhancements at a later date without effecting compatibility. (NB. to do this we must define some limitations on the structure of the future expansion data.) Possible block attributes are shown in appendix 1. An example of how to encode the block attributes is shown in appendix 2.

v. Block Data

[Data]

The exact form of this is for further study.

3. Typical data structure

[PICTURE start code1][PICTURE start code2][PICTURE start code3]
..[GBSC1][GBSC2][GBSC3][BLOCK ADDRESS]....
..[BLOCK TYPE.][BLOCK DATA][BLOCK ADDRESS]..
..[BLOCK TYPE.][BLOCK DATA][GBSC4][GBSC5]..
..[GBSC6]....etc.

4. Generalised VLC

It would be highly desirable to be fixed on one VLC code book which can be used for all situations where VLC occurs. This will both simplify hardware and aid retracking when transmission errors occur.

Appendix 1

1. Picture Attributes

Buffer state
Temporal reference
Use of motion compensation
Intraframe mode
Global motion vectors
Global quantiser stepsize

2. GOB Attributes

GOB number
Global motion vector
Global quantiser stepsize

3. Block Attributes

Fixed/Non fixed block
No MC/MC (+ motion vector)
Non coded/Coded
Interframe/Intraframe
Class number
Quantiser stepsize
EOB (or block length)

Appendix 2

Title: An example of encoding of block attributes

Block type	Attributes (Doc.69) ≠1 (Note 1)	≠2	≠3	≠4	FLC	Example of VLC (Note 2)
1. Fixed	0	-	-	-	000	0
2. Intraframe	1	0	1	0	001	11100
3. Interframe/No MC	1	0	1	1	010	110
4. MC - No coding	1	1	0	-	100+MV	11110+MV
5. MCw/coding	1	1	1	-	101+MV	10+MV
6. MC - No coding + global MV	1	1	0	-	110	11101
7. MCw/coding + global MV	1	1	1	-	111	11111

Note 1: Attributes according to Doc.69

#1 : Fixed/Non Fixed #2 : No MC/MC
#3 : Non coded/Coded #4 : Intra/Inter

Note 2: If relative addressing is used then the first bit of the VLC set may be removed.