CCITT SGXV
Working party XV/1
Specialist Group on Coding
for Visual Telephony

Doc No # 146
November 1986

SOURCE: UK
TILE: A REPORT ON THE MERITS OF USING RELATIVE ADDRESSING FOR BLOCKS WITHIN A GROUP OF BLOCKS (GOB) STRUCTURE.

Two forms of addressing are currently being considered for blocks within a group of blocks (GOB) structure viz: absolute addressing and relative addressing. Absolute addressing has the advantage that the hardware implementation is likely to be simple and the bit usage is not data dependent. The primary disadvantage is that no account is taken of any structure within the blocks being addresssed.

Three relative addressing schemes are examined here. The relative addresses are restricted to within a GOB structure.

In method 1 the relative addresses are generated as indicated in Figure 1. Run lengths are generated for the number of blocks to the next active block.

In method 2 the relative addresses are generated as indicated in Figure 2. Run lengths are generated for the number of blocks to the next fixed block.

In method 3 the relative addresses are generated as indicated in Figure 3. Run lengths are generated for the number of blocks in a cluster of the same type of block.

Note: The results presented here are not in accordance with the recommended methods, as it is thought the presentation here illustrates the results more clearly.

Statistics of relative address usage have been generated using the Okubo Ref Model 2 with the Split Screen/Trevor sequence as the source data. The merits of the three methods are analysed using the VLC table in document #122 (CCITT SGXV Montreal) shown in table 4 therein.

Figures 4 - 12 illustrate histograms for methods 1 - 3. The data is presented for the composite sequence, the Split Screen segment, and the Trevor segment.

While all histograms exhibit similar classes of distribution the occupancy for the relative address value (RAV) of 1 varies significantly. It is this value that appears to affect the relative efficiencies of the different methods. The overall bit rate for the three methods is shown in table 1. This shows the average number of bits used to address each block for the whole of the sequence segment. The bit requirement for the absolute addressing method is indicated in brackets.

METHOD	BITS USED TO ADDRESS THE BLOCKS (ABSOLUTE 1 BIT / BLOCK)	
	SPLIT SCREEN	TREVOR
1	0.821 (1.0)	0.709 (1.0)
2	0.903 (1.0)	0.906 (1.0)
3	0.703 (1.0)	0.622 (1.0)

TABLE 1

Clearly for the source data used here, all methods yield some gain over a 1bit/block scheme. Method 1 appears to be the simplest to implement in hardware but its effectiveness is a strong function of picture activity. If more blocks are active then the population of the RAV 1 is likely to rise and therefore the overall bitrate is likely to increase. This method will operate most efficiently on low activity data and may be found to be very suitable for working at bitrates even lower than 300kbits/s.

The converse is true for method 2. The hardware is likely to be considerably more complex due to the need to buffer coded blocks until an uncoded block is found or alternatively access the buffer somehow to insert appropriate address information. The efficiency of the addressing is likely to increase with picture activity.

Method 3 appears to be the most efficient but probably the most complex to implement in hardware.

The following data applies to results shown in Figures 4 to 13. Each processed frame in the sequence examined had 2376 blocks. The Split Screen sequence had 19 frames and the Trevor sequence had 29 frames.

Comparing the Split Screen segment with the Trevor segment allows one to assess the affect picture activity has on the efficiency of the addressing schemes. Methods 1 and 3 show an improvement in performance for the Trevor sequence compared to the Split Screen sequence whilst method 2 shows a significant degradation. This is as predicted above.

Figure 13 illustrates the histogram for the composite sequence where the coding algorithm is a modified form of the Okubo reference model 2. The modification consists of the introduction of a low pass filter, post motion compensation, applied only to motion compensated blocks. The histogram was generated using method 3. One may see that the distribution is similar to that Figure 12 illustrating some robustness of method 3.

Comparing the results included in this report with previous work, one may further show the robustness of method 3. In previous work an early version of the Okubo reference model 2 was used to generate the results wherein method 2 outperformed method 1 which is the converse of the situation here. Only method 3 performed in a similar manner.

One may conclude that, for the data examined here, Method 3 is the best of the methods examined, but at the expense of the most complex implementation. As the difference in performance between method 1 and method 3 is not too large but the complexity of implementation is significant then it is thought that method 1 is the most suitable for inclusion in any 300kbit/s codec design.

Proposal

Method 1 should be used for any hardware implementation of a codec.

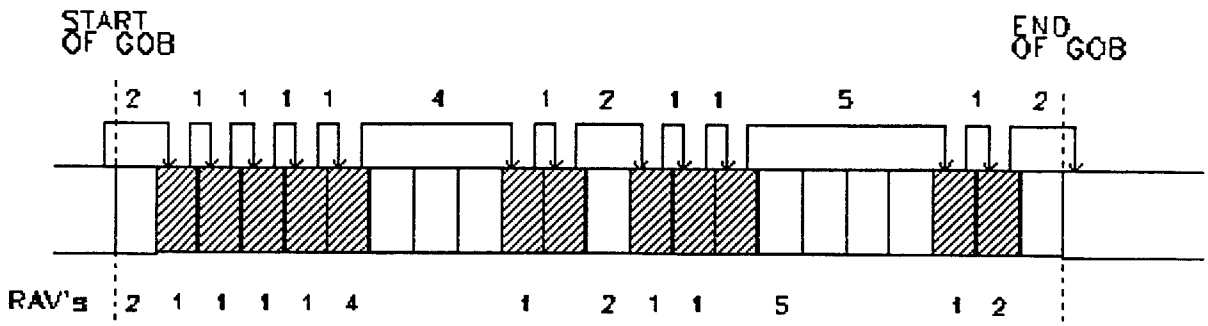


FIGURE 1 – METHOD 1

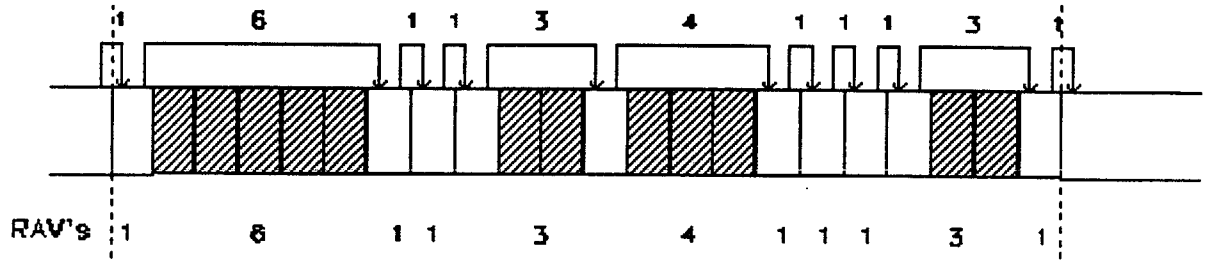


FIGURE 2 – METHOD 2

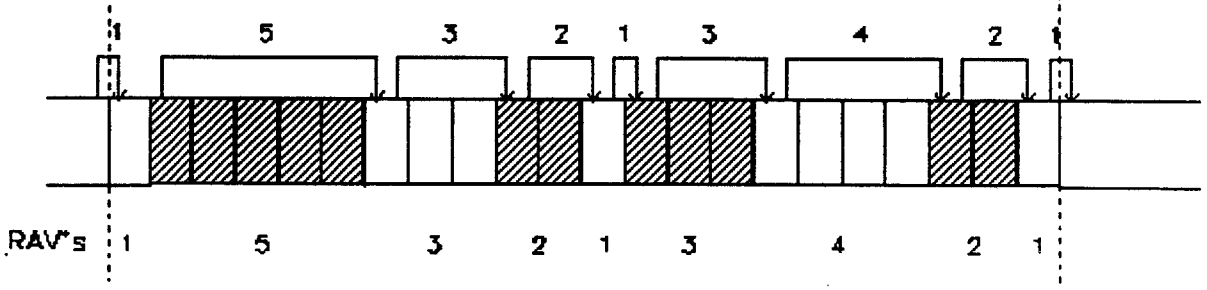


FIGURE 3 – METHOD 3

= ACTIVE BLOCK

= FIXED BLOCK

FIGURE 4

METHOD 1

DATE: 6/10/86
SOURCE: BTRL UK
AUTHOR: G SEXTON

CODER SOFTWARE: CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.

MODIFICATIONS:

SOURCE SEQUENCE: COSTI CIF. SPLIT SCREEN
FRAME SIZE: 352 * 288
BITRATE: 300KBITS/SEC
BUFFER SIZE: 30000

FRAME RATE: 10 FRAMES / S
CODED FRAMES: 1- 19
SCENE CUT: YES. APROIRY KNOWLEDGE.

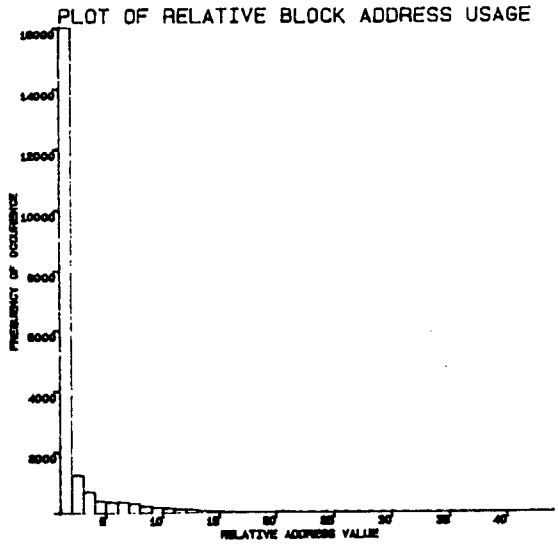


FIGURE 5

METHOD 1

DATE: 6/10/86
SOURCE: BTRL UK
AUTHOR: G SEXTON

CODER SOFTWARE: CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.

MODIFICATIONS:

SOURCE SEQUENCE: COSTI CIF. TREVOR
FRAME SIZE: 352 * 288
BITRATE: 300KBITS/SEC
BUFFER SIZE: 30000

FRAME RATE: 10 FRAMES / S
CODED FRAMES: 20 - 48
SCENE CUT: YES. APROIRY KNOWLEDGE.

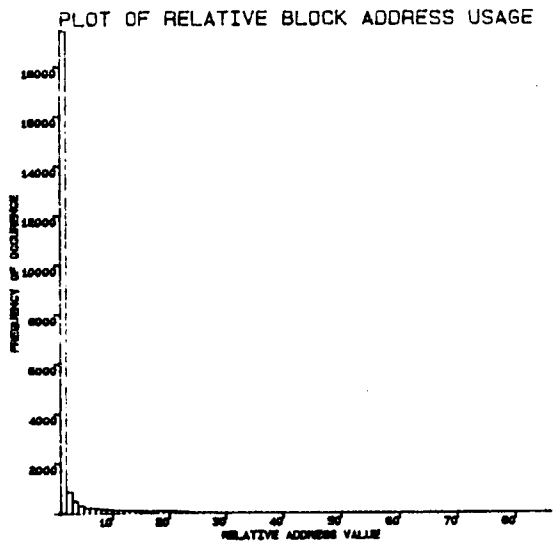


FIGURE 6

METHOD 1

DATE:	6/10/86
SOURCE:	BTRL UK
AUTHOR:	G SEXTON
CODER SOFTWARE:	CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.
MODIFICATIONS:	
SOURCE SEQUENCE:	COSTI CIF. TREVOR/ SPLIT SCREEN
FRAME SIZE:	352 * 288
BITRATE:	300KBITS/SEC
BUFFER SIZE:	30000
FRAME RATE:	10 FRAMES / S
CODED FRAMES:	1 - 48
SCENE CUT:	YES. APROIRY KNOWLEDGE.

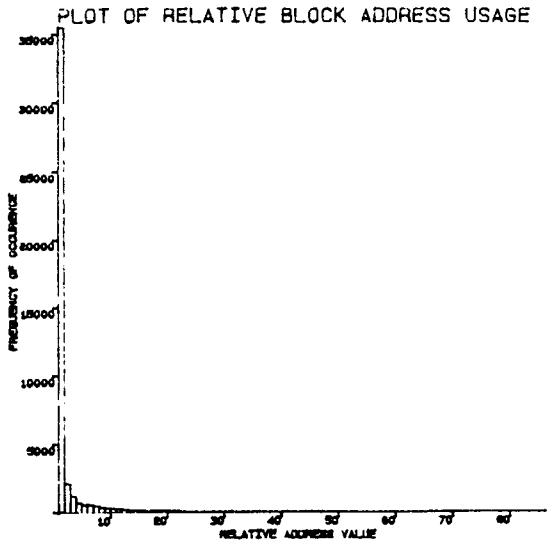
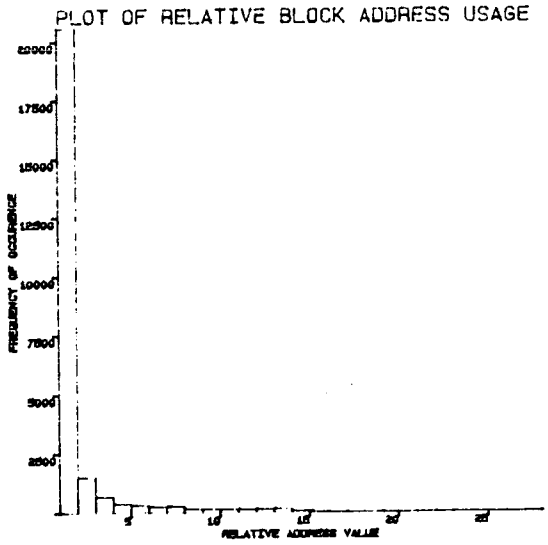


FIGURE 7

METHOD 2

DATE:	6/10/86
SOURCE:	BTRL UK
AUTHOR:	G SEXTON
CODER SOFTWARE:	CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.
MODIFICATIONS:	
SOURCE SEQUENCE:	COSTI CIF. TREVOR
FRAME SIZE:	352 * 288
BITRATE:	300KBITS/SEC
BUFFER SIZE:	30000
FRAME RATE:	10 FRAMES / S
CODED FRAMES:	1 - 19
SCENE CUT:	YES. APROIRY KNOWLEDGE.



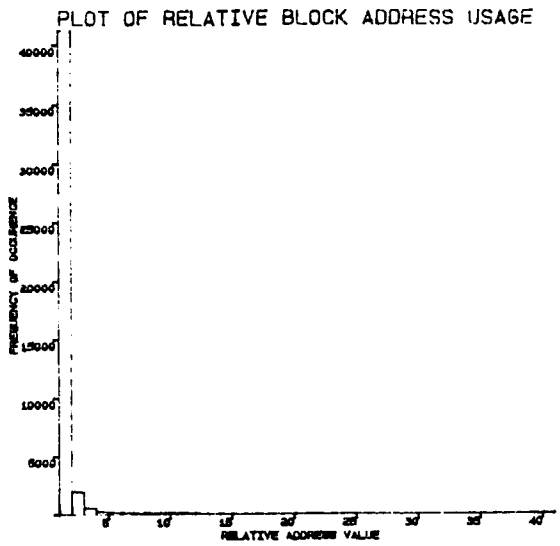


FIGURE 8

METHOD 2

DATE:	6/10/86
SOURCE:	BTRL UK
AUTHOR:	G SEXTON
CODER SOFTWARE:	CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.
MODIFICATIONS:	
SOURCE SEQUENCE:	COSTI CIF. SPLIT SCREEN
FRAME SIZE:	352 * 288
BITRATE:	300KBITS/SEC
BUFFER SIZE:	30000
FRAME RATE:	10 FRAMES / S
CODED FRAMES:	20 - 48
SCENE CUT:	YES. APROIRY KNOWLEDGE.

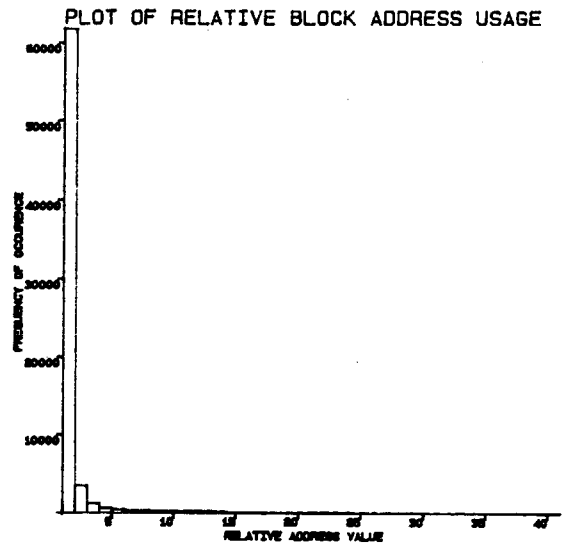


FIGURE 9

METHOD 2

DATE:	6/10/86
SOURCE:	BTRL UK
AUTHOR:	G SEXTON
CODER SOFTWARE:	CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.
MODIFICATIONS:	
SOURCE SEQUENCE:	COSTI CIF. SPLIT SCREEN/ TREVOR
FRAME SIZE:	352 * 288
BITRATE:	300KBITS/SEC
BUFFER SIZE:	30000
FRAME RATE:	10 FRAMES / S
CODED FRAMES:	1 - 48
SCENE CUT:	YES. APROIRY KNOWLEDGE.

FIGURE 10

METHOD 3

DATE: 6/10/86
SOURCE: BTRL UK
AUTHOR: G SEXTON

CODER SOFTWARE: CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.

MODIFICATIONS:

SOURCE SEQUENCE: COSTI CIF. SPLIT SCREEN

FRAME SIZE: 352 * 288
BITRATE: 300KBITS/SEC
BUFFER SIZE: 30000

FRAME RATE: 10 FRAMES / S
CODED FRAMES: 1 - 19
SCENE CUT: YES. APROIRY KNOWLEDGE.

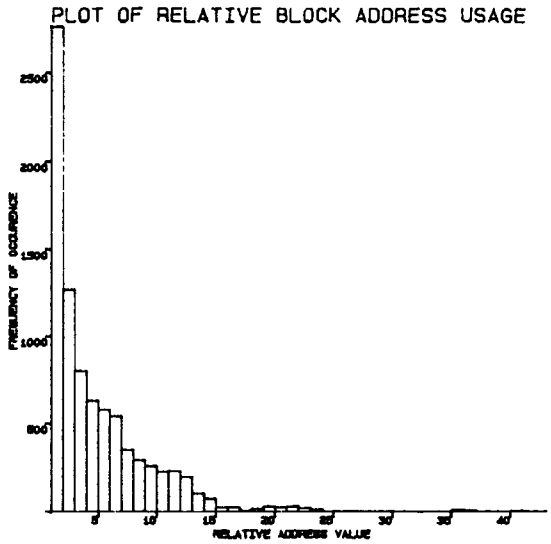


FIGURE 11

METHOD 3

DATE: 6/10/86
SOURCE: BTRL UK
AUTHOR: G SEXTON

CODER SOFTWARE: CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.

MODIFICATIONS:

SOURCE SEQUENCE: COSTI CIF. TREVOR

FRAME SIZE: 352 * 288
BITRATE: 300KBITS/SEC
BUFFER SIZE: 30000

FRAME RATE: 10 FRAMES / S
CODED FRAMES: 20 - 48
SCENE CUT: YES. APROIRY KNOWLEDGE.

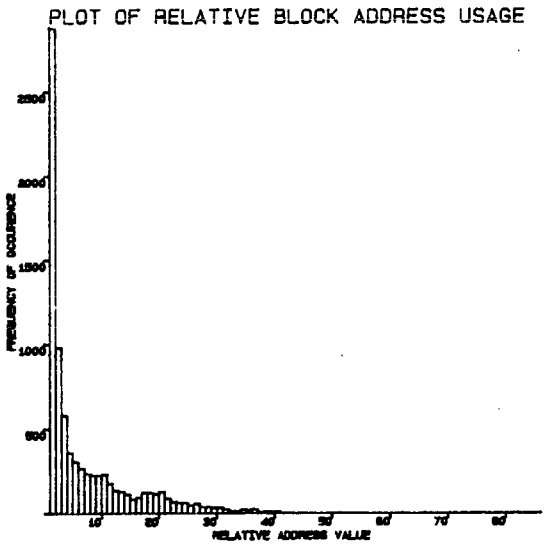


FIGURE 12

METHOD 3

DATE:	6/10/86
SOURCE:	BTRL UK
AUTHOR:	G SEXTON
CODER SOFTWARE:	CODEC3 V1.4 (OKUBO REFERENCE MODEL2 OKUBO8.PRC V1.1.
MODIFICATIONS:	
SOURCE SEQUENCE:	COSTI CIF. SPLIT SCREEN/ TREVOR
FRAME SIZE:	352 * 288
BITRATE:	300KBITS/SEC
BUFFER SIZE:	30000
FRAME RATE:	10 FRAMES / S
CODED FRAMES:	1 - 48
SCENE CUT:	YES. APROIRY KNOWLEDGE.

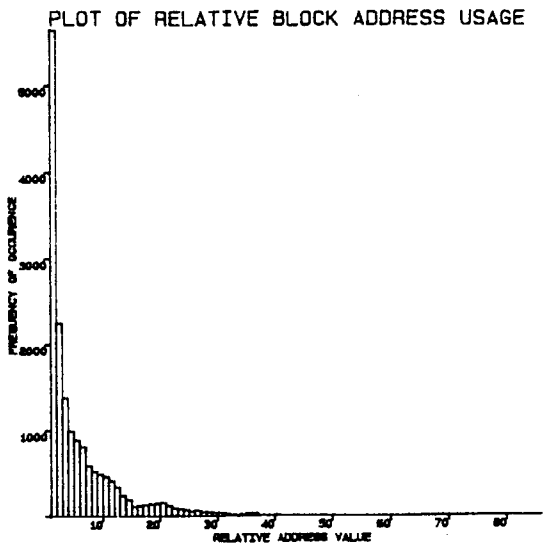


FIGURE 13

METHOD 3

DATE:	6/10/86
SOURCE:	BTRL UK
AUTHOR:	G SEXTON
CODER SOFTWARE:	CODEC5 V1.0 (OKUBO RM2 MODIFIED) OKUBO11.PRC V1.0.
MODIFICATIONS:	LOOP FILTER POST MC
SOURCE SEQUENCE:	COSTI CIF. SPLIT SCREEN/ TREVOR
FRAME SIZE:	352 * 288
BITRATE:	300KBITS/SEC
BUFFER SIZE:	30000
FRAME RATE:	10 FRAMES / S
CODED FRAMES:	1 - 48
SCENE CUT:	YES. APROIRY KNOWLEDGE.