CCITT SGXV
Working Party XV/1
Specialist group on coding for visual telephony
Source: Federal Republic of Germany

Document # 67

January 1986

On Blocksize and Overhead for Displacement Vectors in Block Matching and Object Matching Videotelephone Codecs

Movement compensated prediction and interpolation is a key element of source coding of television signals down to some kbit/s for bidirectional videoconference or videotelephone applications. A comparison between block matching and object matching techniques is made in this paper with special regard to the choice of blocksize and the resulting overhead needed as a description of the movement vectors.

Two objective figures are used for the comparison:

a) The gain by movement compensation (mean square deviation).

b) The amount of overhead bits for the displacement vectors.

One subjective figure will be considered:

c) introduced artefacts in the reconstructed pictures.

Figure 1 shows a basic block diagram of the coder and decoder with movement compensation. The structure of this coder is identical to the agreed proposal of the COST 211 bis simulation subgroup. At the input side after the scan conversion (SC) there is the motion analyzer MA determining displacement vectors from the original pictures O_m and O_{m-n} (m is the frame number and the delay time of n*t corresponds to the delay time within the DPCM loop, n is the number of skipped fields +1 and t is the field period). The estimator E is controlled by the output signal of MA and calculates an estimated picture S_m for prediction from the previous recostructed picture R_{m-n}. In the DPCM loop the residual prediction errors are transformed (DCT), quantized (Q) and coded for transmission, also the inverse transform (IDCT) is performed in order to generate the recontructed picture in the time domain. After variable length coding (VLC) and buffering (B) the quantized prediction error signal, the classification map and control bits of the quantizer and the overhead information for the changed areas (moved blocks or objects) with their corresponding movement vectors are multiplexed (MUX) before transmission. Displacement vectors are only calculated from the luminance signal, the chrominance vectors are extrapolated luminance vectors.

The decoder performs basically inverse operations. In order to achieve smooth movements skipped fields are interpolated motion-adaptively by evaluating the transmitted displacement parameters. This involves the necessity of measuring "true" displacements, not only minimizing the prediction error or some other cost function. The used technique of movement compensation and interpolation by object matching and the quantizing strategy are explained in detail in /1/. A technique for motion-adaptive interpolation measuring "true" displacements at the receiver side without using the transmitted vectors is described in /2/.

/1/ G. Kummerfeldt, F. May, W. Wolf
Coding Television signals at 320 and 64 kbit/s
2nd International Technical Symposium on Optical and Electro-Optical Science and Engineering, SPIE Conf. B594 'Image Coding', Cannes, France, Dec. 1985

/2/ R. Thoma
A Refined Structure of a Motion Compensating Interpolation Algorithm
Picture Coding Symposium 86, April 1986, Tokyo, Japan

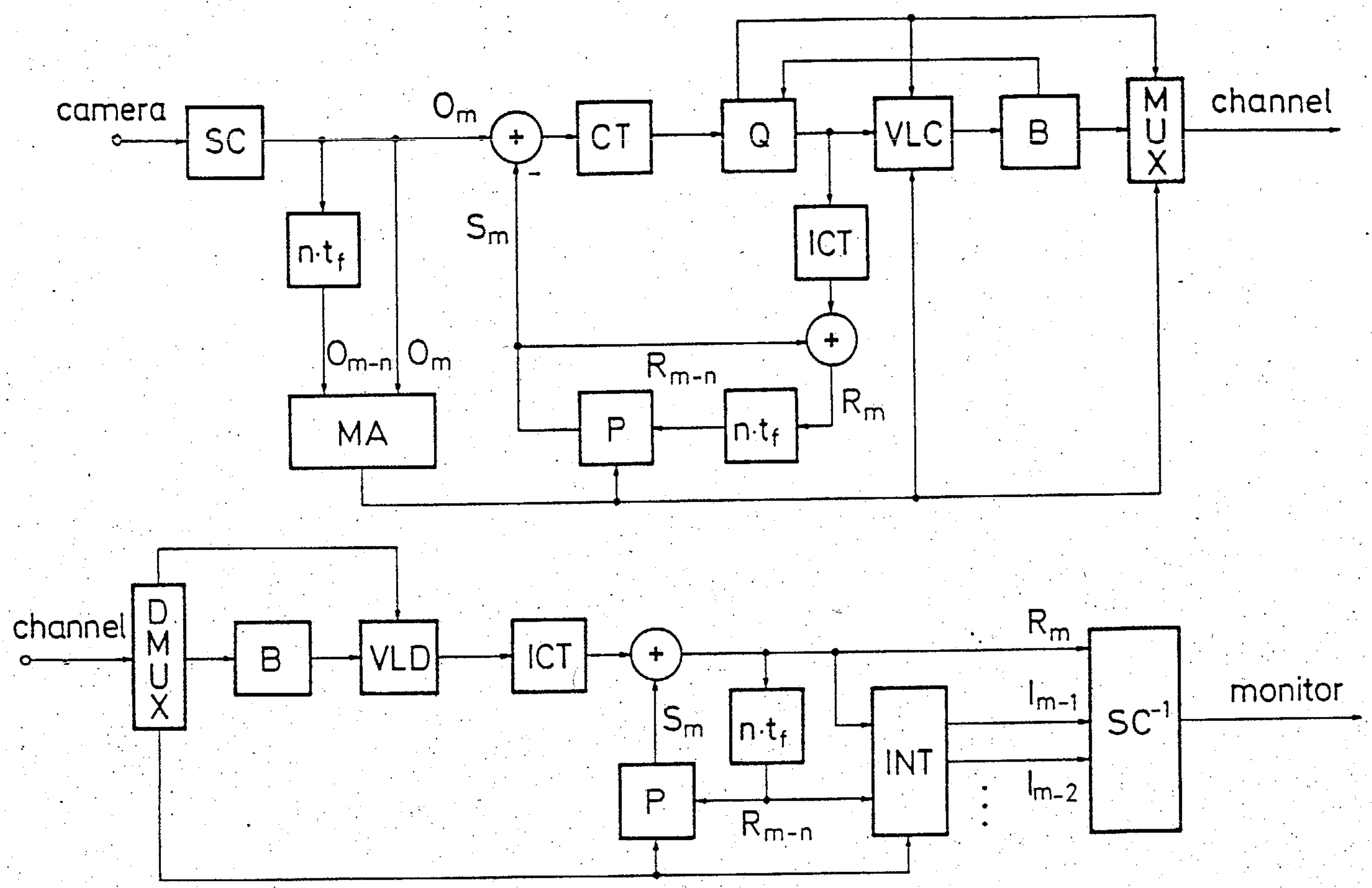

Figure 1. Structure of motion compensated coder and decoder

For the investigations described here we reduced the number of luminance pixels per active line to 256 and the number of chrominance pixels per active line to 64. The luminance signal has 288 lines and the chroma signals U, V have 144 lines each. The field frequency is reduced to 12.5 Hz. The horizontal resolution is somewhat lower than the agreed resolutions within COST 211 bis and Study Group XV of CCITT, but this doesn't affect the comparison aimed at. Numerical results valid for the higher resolution can be calculated from the figures to be presented.

Movement compensation by block matching is performed by subdividing the fields to be coded in n*n blocks (here n=8 and n=16 are considered). For each block a displacement vector is calculated and, if this vector is unequal zero, this vector has to be coded and transmitted. For the comparison made displacement vectors are found by searching the minimum of the absolute value distance function (window = n*n, full search).

Object matching can be considered as an extension of block matching. Adjacent changed blocks (also 8*8 and 16*16 are considered) are concatenated to changed areas, which, under ideal conditions (stationary background, no occlusions, constant illumination) represent one moving object. For this object only one representative movement vector is calculated based on a movement model and transmitted. The non ideal behaviour of natural scenes led us to calculate one or two models in order to solve problems of occlusions and changing shape.

For comparison we used the COST sequences 'Split', 'Trevor' and 'Cut' as scene material. First, the transfer function of the quantizer has been set to one in order to study the gain of both movement compensation techniques and the necessary overhead bits for the displacements vectors without any influence from the quantizing procedure.

In the following Figures 2 through 8 the inverse normalized mean square deviation has been drawn (in db), which can be considered as a signal to noise ratio. Curve a) corresponds to the deviation between the actual field to be coded and the preceding field without motion compensation, curve b) to the deviation between the estimated picture when motion compensation has been used. In Figure 6,7 and 8, curve c) shows the deviation between the original and the reconstructed pictures when quantizing down to a fixed rate of 256 kbit/s has been used. The used formula with summing over the whole picture area is:

$$S/N = -10*\log \frac{\sum\sum (O_m - R_m)^2}{\sum\sum (O_m)^2}$$

For curve a) the expression in the nominator is $(O_m - R_{m-n})$, for curve b) $(O_m - S_m)$.

Figure 2 shows the results of block matching whith 8*8 blocks, Figure 3 whith 16*16 blocks. Figure 4 shows the results for object matching when the boundaries of the objects are approximated by 8*8 blocks, and Figure 5 by 16*16 blocks.

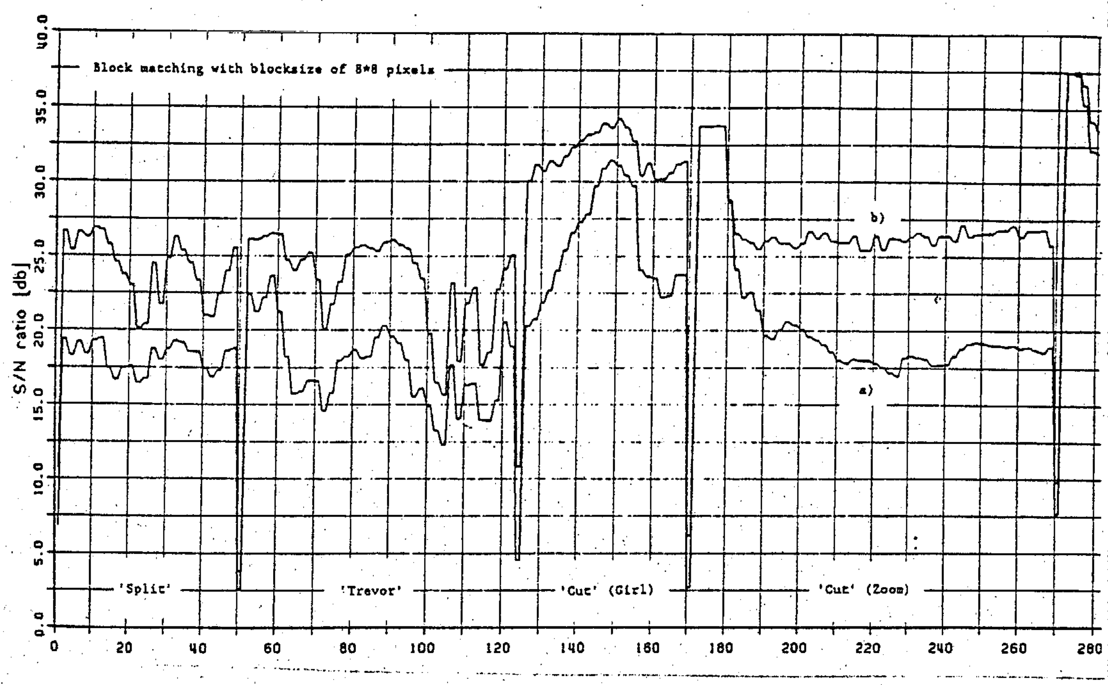

Figure 2.: Block Matching with blocksize 8*8

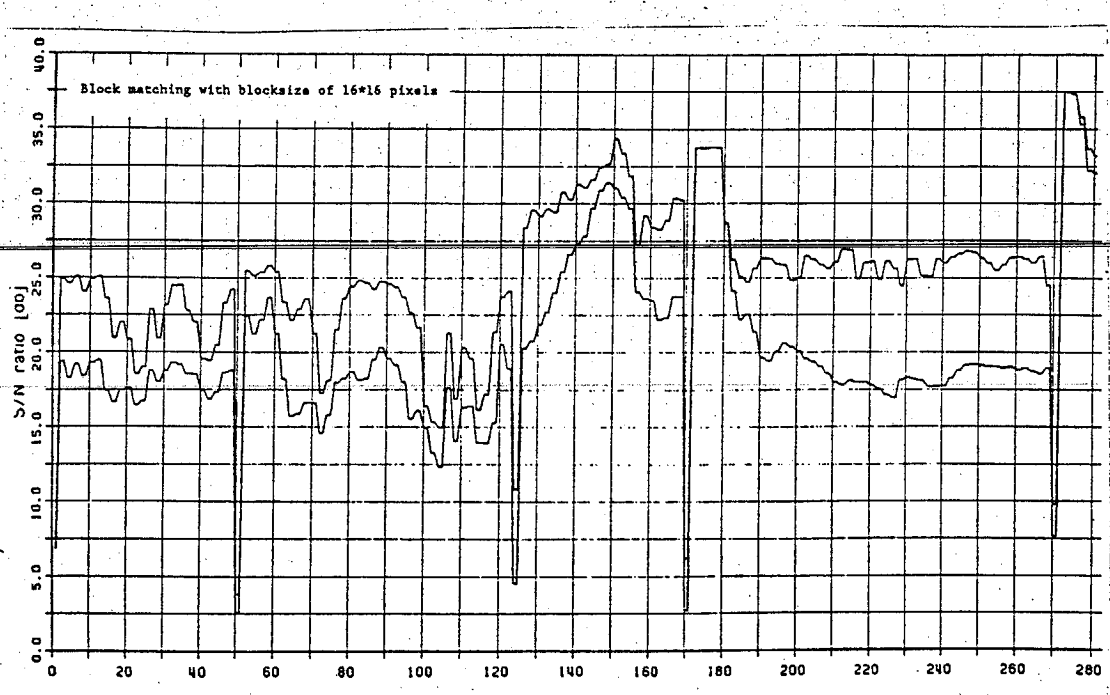

Figure 3.: Block Matching with blocksize 16*16

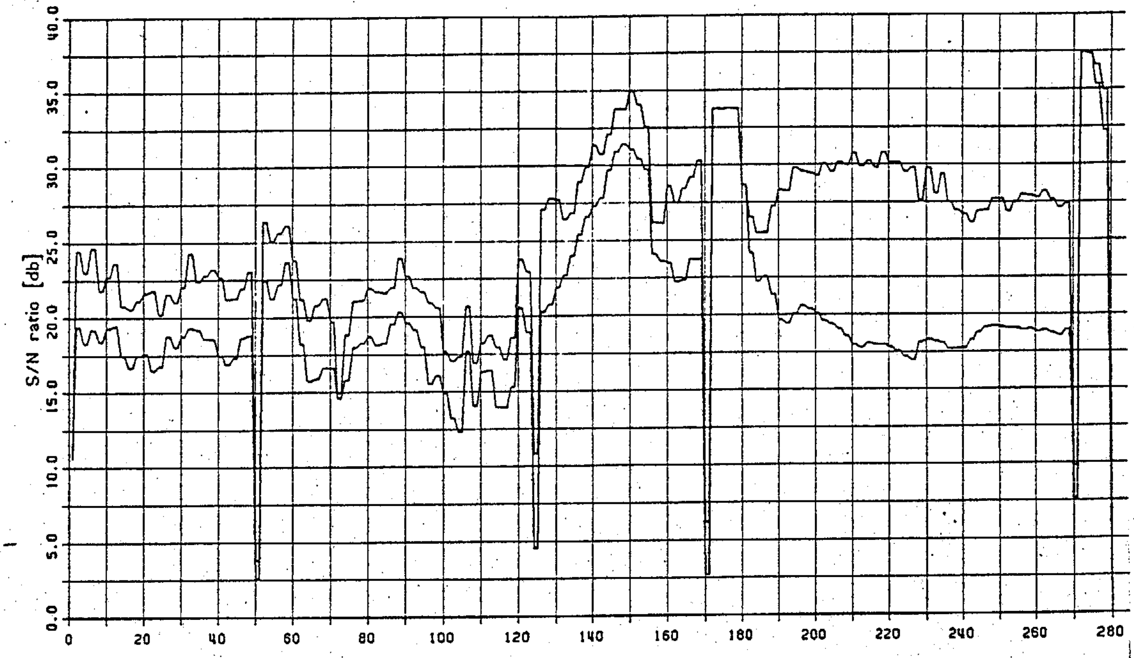

Figure 4.: Object Matching based on blocksize 8*8

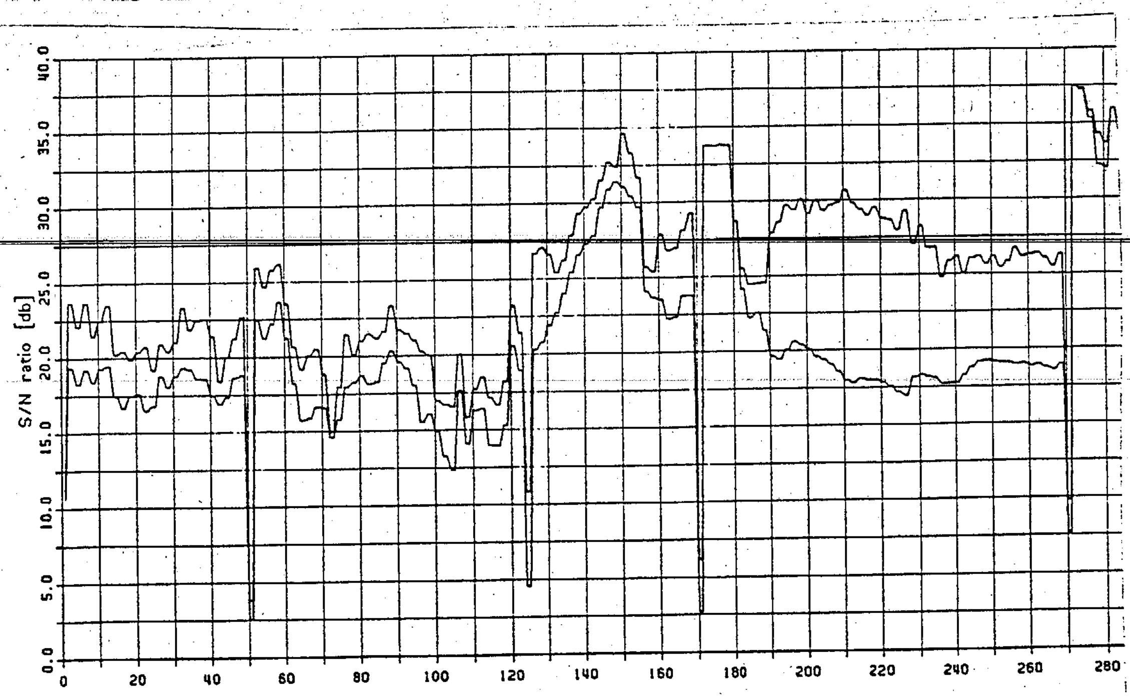

Figure 5.: Object Matching based on blocksize 16*16

The results shown in Figure 2 through 5 are not surprising. It's quite clear that block matching with 8*8 blocks yields a higher gain than with 16*16 blocks due to the fact of a more precise circumscribing of the moving areas. On the other hand, block matching with 16*16 block needs approximately one third of the amount of overhead bits as compared to 8*8 blocks.

It's also quite clear that object matching with three dimensional translation as the movement model results in a lower gain when coding human people with changing shape. We are working on a better segmentation and better suited movement models. But the curves show clearly, that the used movement model of 3-D translation is valid in the zoom and pan part of the Cut sequence (field number 180-270), resulting in a significantly higher gain. This is mostly due to the fact that object matching is able to eliminate wrong measurements and allows the transmission of non-integer representative movement vectors.

For block matching the amount of overhead bits has been calculated as follows: 1 bit per block moved-unmoved discrimination, 2*4 bits per moved block for the two (integer) components of the displacement vector (this allows to code displacements from -8...+8 pixels). The luminance signal resolution mentioned above gives theoretical maximum values (in the case of a completely moved field) of 2592 bits for 16*16 blocks and of 10368 bits for 8*8 blocks. Using the correct COST format these figures will be 3645 and 14580 bits. Having in mind the amount of bits per coded field allowed (some 25000 bits when coding at 320 kbit/s), a look these figures is very interesting.

The calculation of overhead bits for object matching is somewhat more complicated and explained in detail in /1/. Basically, the boundaries (approximated by 8*8 or 16*16 blocks) of the moving objects and the (non integer) components of the movement vector, now representative for one changed area and not for one block, have to be transmitted . With a limitation to 8 moving objects and two parameter sets for the movement model (in order to solve the problem of occlusions) per moving area, the theoretical maximum amount of bits is 1344 when the contours are approximated by 16*16 blocks and 3648 bits when using 8*8 approximation. These figures are significantly lower than the block matching figures. On the other hand, as mentioned above in the case of objects with changing shape the gain by movement compensation will be lower. The following table shows measured mean values and maximum values of overhead bits for the COST sequences.

	Split mean/max	Trevor mean/max	Cut mean/max
block matching 8*8	3499/3852	3319/4020	3718/5928
block matching 16*16	1032/1128	973/1216	1200/2040
object matching 8*8	1500/1864	1331/1936	1549/2970
object matching 16*16	559/781	384/554	448/806

Table 1.: Overhead bits for movement compensation

The results concerning the gain would imply a preference for block matching. But looking at S/N figures of reconstructed pictures when coding the scenes at a fixed rate of 256 kbit/s, another answer will be found.

First, the big amount of bits needed for transmission of the displacement vectors is no more available for coding the residual prediction errors. Second, and more important, block matching works with integer displacement vectors; this implies the introduction of artefacts at the blockborders in the estimated picture and, due to the limited amount of bits available, in the reconstructed picture. These artefacts (mostly short horizontally and vertically oriented line pieces) are distributed all over the moving area. The principle of object matching allows such artefacts only at the borders of the moving area, not inside. Experiments have shown, coding with different blocksize for movement compensation and for block quantizing (e.g. 8*8 for block matching, 16*16 for transform coding) gives worse results. Object matching allows different blocksizes. Nevertheless we found that object matching with 16*16 approximation of the contour of moving areas and 16*16 block quantizing gave the best subjective impression of the reconstructed pictures. Also block matching and block quantizing with common blocksize 16*16 is superior to the common blocksize 8*8.

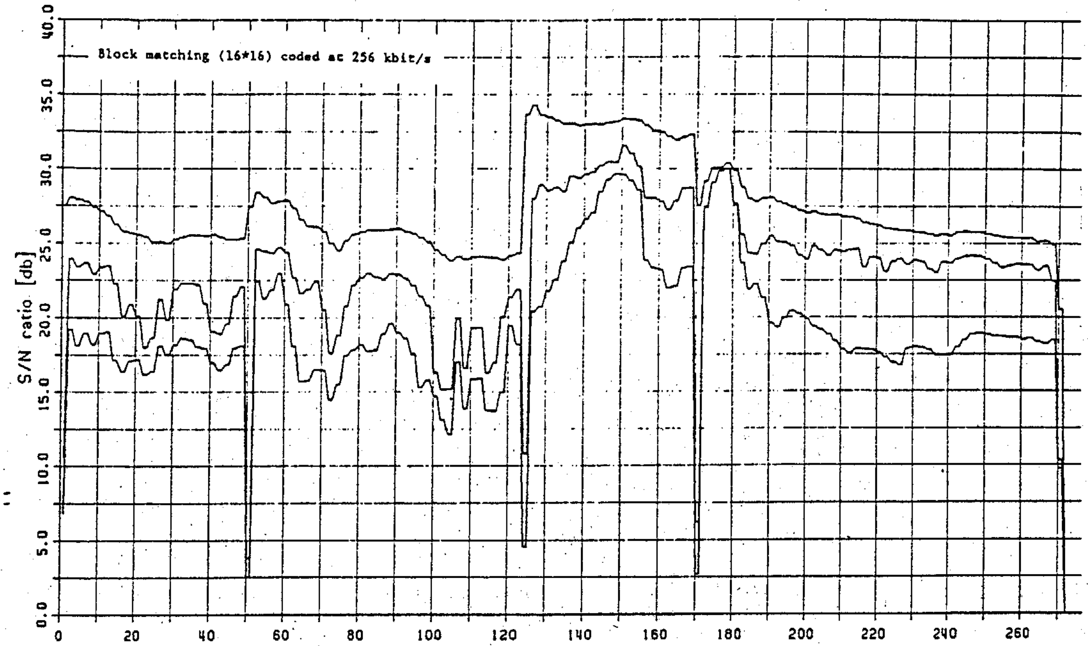

Figure 6.: Coding at 256 kbit/s with Block Matching (16*16)

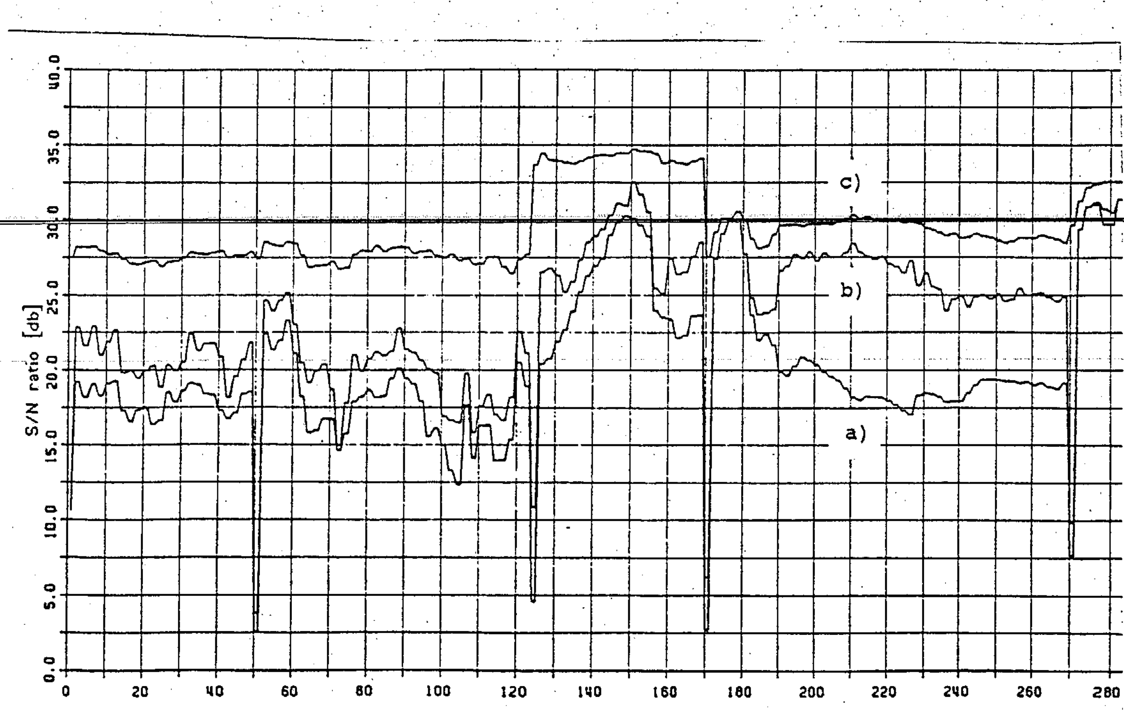

Figure 7.: Coding at 256 kbit/s with Object Matching (16*16)

The accumulation of the artefacts mentioned above causes the decrease in S/N ratio after each scene change as shown in Figure 6 c). We tried to reduce this annoying effect and eliminated the discontinuity of the displacement vectors at the block borders by interpolating the vectors bilinearily, resulting in non-integer displacements; this implies also bilinear interpolation of the amplitudes when calculating the estimated picture, which increases the amount of calculations needed by 14 operations per moved pixel. Figure 8 shows, that the gain by movement compensation is somewhat higher and the artefacts are reduced (not eliminated); nevertheless object matching yields better S/N ratio and a better subjective picture quality.

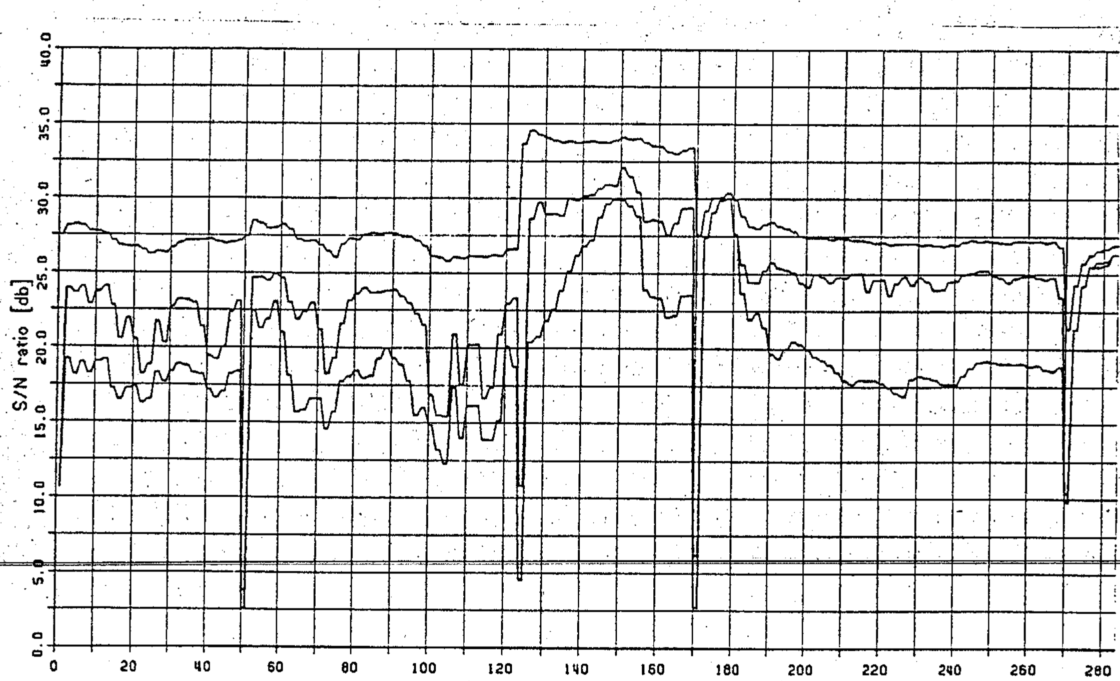

Figure 8.: Coding at 256 kbit/s with Block Matching (16*16) with interpolated non-integer displacement vectors

Conclusions:

Compared to object matching, block matching gives higher gains by movement compensation when coding objects with changing shape. On the other hand, a significantly higher amount of bits is needed for the transmission of the displacement vectors, this amount beeing no moore available for the coding of the residual prediction errors. In the case of changing camera parameters (zoom, pan), the gain of object matching is higher with a lower overhead needed. With object matching "true" displacements are estimated. This allows the use of the transmitted displacement vectors for motion adaptive interpolation of skipped fields; nevertheless there is the possibility to improve the performance using a more sophisticated post-processing in order to display smooth movements on the receiver side.

A coding scheme with block matching and block quantizing should use the same blocksize for movement compensation and quantizing, and a bilinear interpolating calculation of the estimated picture gives better picture quality. Otherwise high frequency artefacts are introduced when the transmission rate is limited. These artefacts are considerably lower with object matching.

In our experiments a blocksize of 16*16 for both movement compensation and quantizing yielded the best subjective picture quality. If a bitrate of 64 kbit/s is aimed at, the amount of 7680 bits per coded field (field frequency 8.1/3 Hz) does not allow a blocksize of 8*8.

The results presented make us recommend for standardization a blocksize of 16*16 both for block matching and block quantizing. The possibility of decoding signals with movement compensation by object matching should be included in a future standard.