Block-Classified Motion Compensation Scheme for Digital Video*

Sohail Zafar† Member, IEEE **Ya-Qin Zhang**‡ Senior Member, IEEE
Bijan Jabbari§ Senior Member, IEEE

Abstract

A novel scheme for block-based motion compensation is introduced in which a block is classified according to the energy that is directly related to the motion activity it represents. This classification allows more flexibility in controlling the bit rate and the signal-to-noise ratio and results in a reduction in motion search complexity.

The method introduced is not dependent on the particular type of motion search algorithm implemented and can thus be used with any method assuming that the underlying matching criteria used is minimum absolute difference. It has been shown that the method is superior to a simple motion compensation algorithm where all blocks are motion compensated regardless of the energy resulting after the displaced difference.

I Introduction

One reason that motion compensation provides such promising results is that the majority of the motion in a natural scene is translatory. Motion compensation tries to estimate, for a pixel or for a block in the current frame, the position in the previous frame [1, 2, 5, 6]. The most popular matching methods appearing in the literature are maximum cross correlation (MCC), minimum mean squared error

*This work was partially supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

†Sohail Zafar is with the Mathematics and Computer Science Division at Argonne National Laboratory, Argonne, IL. (E-mail: zafar@mcs.anl.gov)

‡Ya-Qin Zhang is with David Sarnoff Research Center, Princeton, NJ. (E-mail: zhang@earth.sarnoff.com)

§Bijan Jabbari is with the Electrical and Computer Engineering Department at George Mason University, Fairfax, VA. (E-mail: bjabbari@gmu.edu)

(MMSE) or minimum absolute difference (MAD). All the methods find the position where the function they represent is minimum or maximum.

Minimum absolute difference, which is most widely used because of low computational complexity, minimizes the sum of the absolute differences of all the pixels in that block. It is given by

$$\mathbf{V}_i(x,y) = \arg\min_{x,y\in\Omega}\Big\{\frac{1}{XY}\sum_{p=-X/2}^{X/2}\sum_{q=-Y/2}^{Y/2}\big|I_i(x_m+p,y_n+q) \\ - I_{i-1}(x_m+p+x,y_n+q+y)\big|\Big\}, \quad (1)$$

where $\mathbf{V}_i(x,y)$ is the motion vector, $I_i(.,.)$ is the pixel value in the current frame, and $I_{i-1}(.,.)$ is the value in the previous frame. The range of x and y is called the search area and is denoted by $\mathbf{\Omega}$.

Motion estimation seeks to find a position in the previous frame, such that only the minimum amount of information (difference of the two corresponding pixels) must be sent. In areas of very little or no motion activity, the difference image without any motion compensation will be very small, and motion compensation may not result in a significant gain in terms of bit rate or signal-to-noise ratio. Thus, it is better not to waste any computational effort calculating motion vectors for such blocks.

Similarly, when occluded areas in a scene are uncovered because of a moving object, that area has no correspondence in the previous frame. Such blocks, if transmitted without being compensated for motion, have less energy than if they are displaced to any position and subtracted from a block in the previous frame, because the sum of the absolute values of the pixels in the blocks itself may be lower than the sum of the minimum computed by MAD ($|x|$ can be less than $|x-y|$). In this case, it is better to transmit the block "as is" rather than as a difference. Finally, the motion compensation scheme may fail to track the object because of insufficient search area. In such a case it is better to transmit the block as is rather than forcing it as a difference with a nonrelated block. This is the essence of block classified motion compensation scheme.

In the proposed block classification scheme every block is classified according to the motion activity that block represents [4, 8]. The amount of motion is reflected by the amount of energy the block contains. The advantages of using such a scheme are fourfold. First, it decreases the search complexity, since no motion vectors are searched for the blocks that do not require compensation. Second, it reduces the motion overhead, since information needs to be sent only for blocks that are motion compensable. Third, each type of block can be treated with different priority in terms of quantization. Blocks that are uncompensable contain the most energy, since they belong to freshly uncovered regions and can be quantized with more bits than those that are compensable. The blocks that do not require any compensation may be quantized with the least amount of bits, thus saving in overall bit rate. The biggest advantage this scheme has over others is its ability to automatically handle abrupt scene cuts because almost all the blocks will be automatically classified uncompensable and that frame will be handled just like a refresh frame. The only disadvantage is that the information about the type of block must be transmitted, thereby adding to the bit rate.

II Classification Algorithm

The first step of the algorithm is to calculate a quantity that is a measure of the energy of the block in question without motion compensation. Calculating the energy directly is computationally expensive, so the following measure is calculated which requires only comparisons and additions. Let the pixel value at any location (x_m, y_n) within the block in question for frame i be denoted by $I_i(x_m, y_n)$ and by $I_{i-1}(x_m, y_n)$ for the corresponding pixel in the previous frame $(i-1)$. We compute $|\mathcal{S}_\infty|$, the cardinality of the set $\mathcal{S}_\infty$, which is defined as

$$\mathcal{S}_\infty = \{I_i(x_m, y_n) : |I_i(x_m, y_n) - I_{i-1}(x_m, y_n)| > \theta_1 \quad \forall \quad 0 \leq m \leq X, 0 \leq n \leq Y\}, \quad (2)$$

where X and Y are the dimensions of the block. In simple words, $|\mathcal{S}_\infty|$ denotes the number of pixels in the non-motion compensated difference block having a value greater than θ_1. So, if $|\mathcal{S}_\infty| < \phi_1$ (i.e., there are fewer than ϕ_1 pixels in the block having a value higher than θ_1), the block certainly belongs to an area with very low motion activity and does not need to be motion compensated. If the number of pixels having a value above θ_1 is greater than ϕ_1, the block is a candidate for motion compensation.

Once it has been determined that the block can be compensated because it contains fair amount of energy with simple difference, any motion-searching criterion can be used to find the motion vectors (full search, predictive search, etc.). After the motion vectors are found having a value of (x, y), the compensated difference block is again tested for energy. We compute $|\mathcal{S}_\in|$, where $\mathcal{S}_\in$ is defined as

$$\mathcal{S}_\in = \{I_i(x_m, y_n) : |I_i(x_m, y_n) - I_{i-1}(x_m + x, y_n + y)| > \theta_2 \quad \forall \quad 0 \leq m \leq X, 0 \leq n \leq Y\}. \quad (3)$$

In other words, we find number of pixels in the block whose motion-compensated difference is higher than a value θ_2. If $|\mathcal{S}_\in| < \phi_2$ (i.e., the number of pixels in the

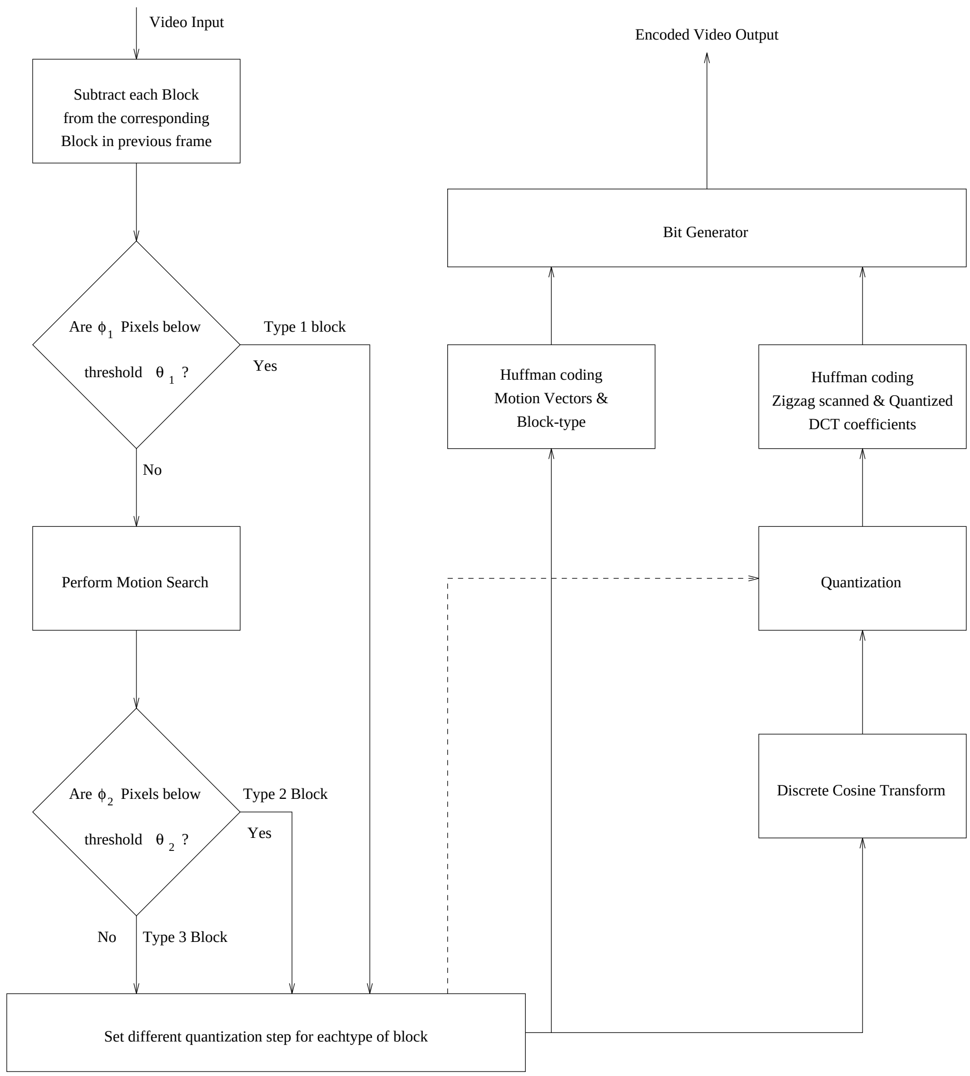


Figure 1: Block Classification Algorithm

compensated block having a value higher than θ_2 is less than ϕ_2), the block is motion compensable. A flow diagram of the algorithm is shown in Figure 1.

If $|\mathcal{S}_\in| > \phi_2$ the block is termed as noncompensable because the energy in the compensated block is quite high and it is better to transmit the block as is, without taking any difference. Therefore, a block is labeled as type *1*, type *2*, or type *3* if it satisfies the following:

$$T = \begin{cases} \mathit{1} & \text{if } |\mathcal{S}_\infty| < \phi_1, \\ \mathit{2} & \text{if } |\mathcal{S}_\infty| > \phi_1 \text{ and } |\mathcal{S}_\in| < \phi_2, \\ \mathit{3} & \text{if } |\mathcal{S}_\in| > \phi_2. \end{cases} \tag{4}$$

This classification information is then used by the quantizers and is also transmitted as an overhead.

III Description of the Codec

The video codec implemented in the simulations is very similar to the one described under the general MPEG standards [3] and is shown in Figure 2. The main difference is that the block-type information is supplied to the quantizers and the frame buffer and is also sent to an entropy coder for transmission.

The first step for nonrefresh frames is block classification (Equation 4) which also calculates the motion vectors in the process. After the blocks are classified as either type *1*, type *2*, or type *3* by the algorithm, the displaced frame difference is computed, and quantization is performed based on this information. The quantization tables used in the simulation for both the luminance and chrominance part are the same as those defined in the JPEG or MPEG standards [3, 7].

Type *1* blocks are quantized with the fewest number of bits (coarse quantized with a very high scaling factor), since they contain very small energy. Type *2* blocks, which are the normal motion-compensated blocks, are quantized with a smaller

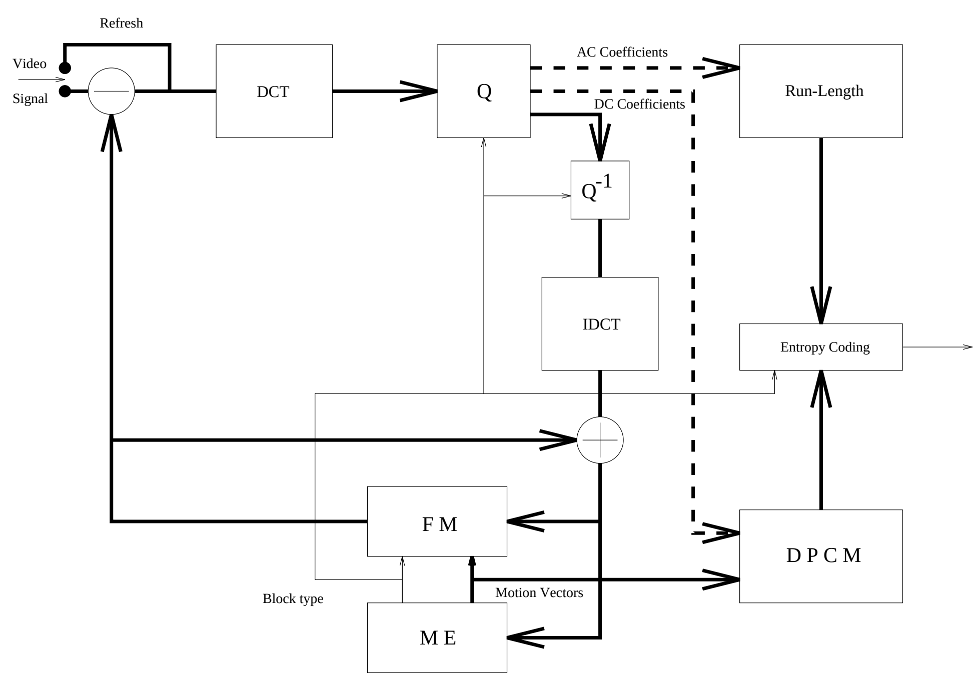


Figure 2: Block Diagram of the Motion-Classified Block DCT Video Codec

scaling factor (quantization levels) than that of type *1* blocks. The remaining blocks, which are of type *3* (i.e., they are not block differences but blocks which are to be transmitted as is), are quantized with a much smaller scaling factor. It should be noted here that in the refresh frames all the blocks are of type *3*, since the refresh frames are sent without any interframe compensation.

The motion vectors and the block-type information are transmitted without any quantization. The first order entropy is calculated to find the overhead. As mentioned before, no attempt has been made to efficiently code the motion and the block-type information in the simulation results that follow. Such an effort will further decrease the number of overhead bits, but the entropy gives the theoretic upper limit.

IV Simulation Results

The "car" sequence was used in the simulations of the above-mentioned codec. Figure 3 shows the histogram of the three types of blocks for a typical frame. It can

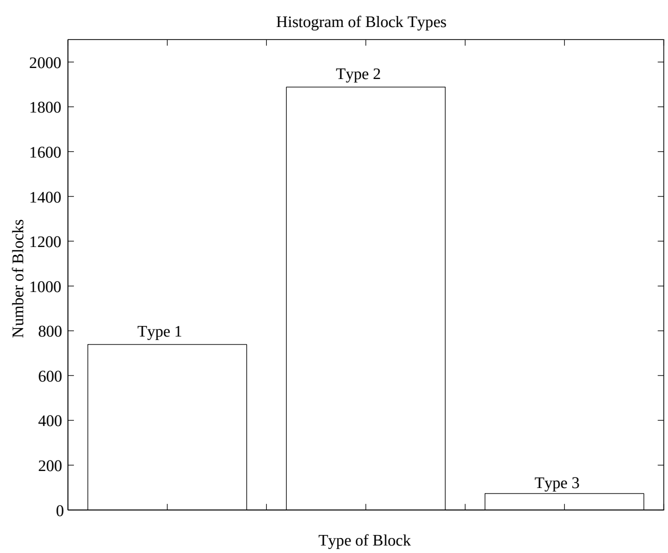


Figure 3: Histogram of Block Type for a typical frame

be seen that even for this particular sequence, which is a panning camera following a car, almost 27% of the blocks are type *1* and do not require any motion compensation. Thus, the search complexity and motion overhead are reduced by the same amount. It should be noted that this reduction is independent of the type of search implemented. Furthermore, this percentage is highly dependent on the motion at that particular instant. In the worst-case scenario (when there are no blocks of type *1*), this scheme will perform exactly the same as the one that does not employ block classification in terms of signal-to-noise ratio, but it will have a slightly higher bit rate because of added block-type overhead. Observation and simulation results show that this scenario never happens in real life (for more details refer to [4]).

Figure 4 shows the instantaneous bit rate compared with a full motion scheme with equivalent parameters. FS24 represents the results for a normal motion-

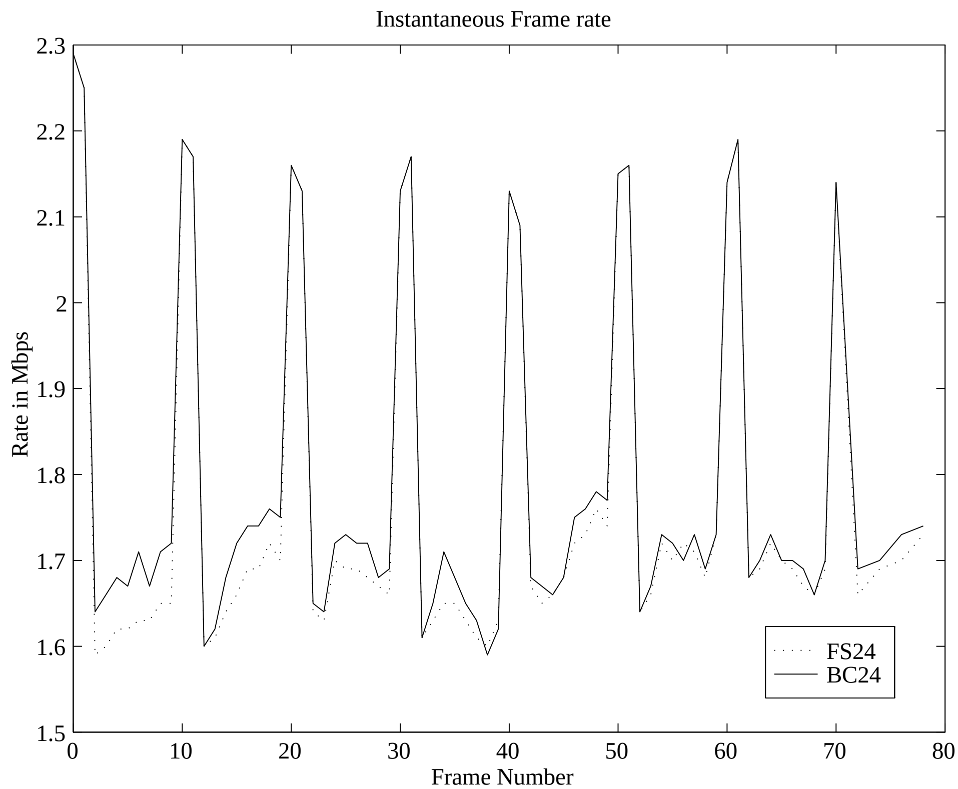


Figure 4: Instantaneous Bit Rate

compensated scheme without block classification, whereas BC24 represents a block-classified scheme having a search area of ± 24 pixels in both the horizontal and the vertical direction. In this particular simulation, $\theta_1 = 5$, $\phi_1 = 16$, $\theta_2 = 8$, and $\phi_2 = 32$. In the initial block difference, if fewer than 16 pixels have an absolute value greater than 5, the block is labeled as type *1*, and no motion compensation is performed. If more than a quarter of the pixels (since the block size is 8×8) have a value greater than 5, the block is a suitable candidate for motion compensation. A full-search scheme is used in this section. After the motion vectors have been calculated, the displaced difference is again tested for energy with $\theta_2 = 8$ and $\phi_2 = 32$. If half of the pixels have an absolute value greater than 8, the block is classified as noncompensable (type *3*) and transmitted as is; otherwise, the motion vectors are accepted and the block labeled as type *2*. It can be seen from the figure that the block-classified scheme has a bit rate that is almost the same as the full search without block classification. The increase in bit rate from block-type overhead and more bits required for type *1* and *3* blocks is compensated by the decrease in motion overhead.

Figure 5 shows the motion, block-type overhead and also the total overhead for the block-classified scheme in comparison with a standard motion compensation algorithm. It clearly shows that the total overhead for block-classified scheme is re-

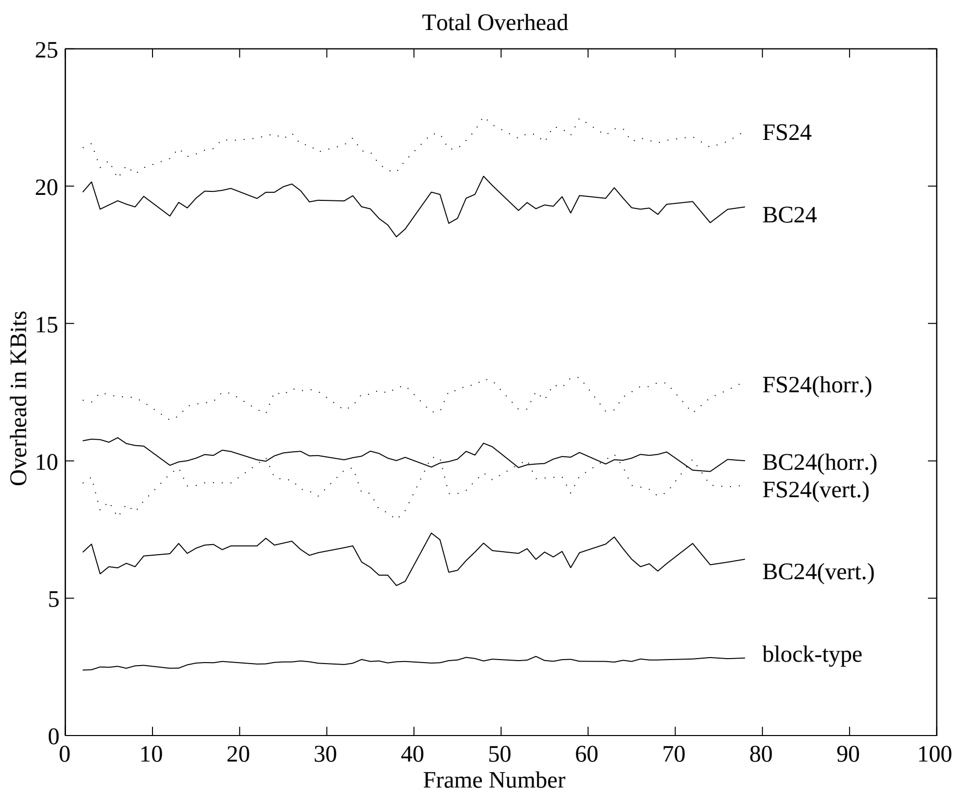


Figure 5: Motion and Block-type Overhead

duced despite the fact that block-type information has to be transmitted in addition to the motion vectors. This reduction results because there are only three types of blocks; thus the maximum entropy is $\log_2 3$, and this is the only information needed to be sent for type *1* and *3* blocks. The maximum entropy for the motion vectors is $\log_2(2p+1)$ with a search area of $\pm p$, which has a value of 24 in this case.

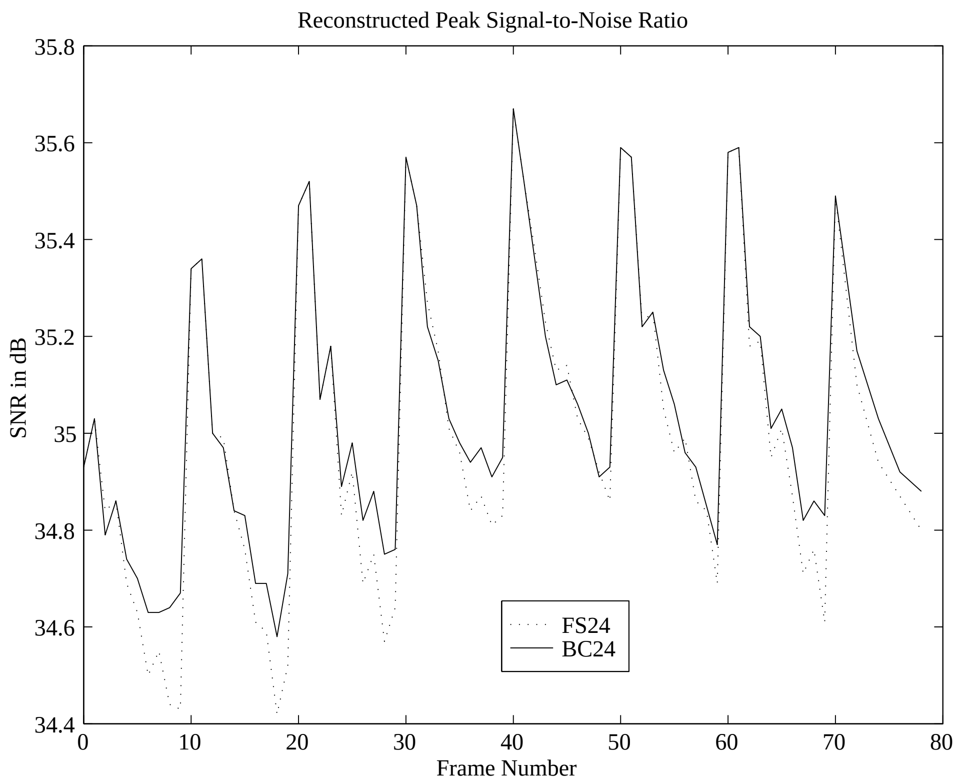


Figure 6: Reconstructed Peak Signal-to-Noise Ratio

Figure 6 shows the signal-to-noise ratio of the block classified motion compensation scheme in comparison with a normal full search. The graphs shows a slight improvement in the signal-to-noise ratio when using block classification. Though the improvement is not substantial, the reduction in search complexity is significant. It should be noted that this scheme provides a better signal-to-noise ratio *vs* bit rate control than one without block classification. The parameters of both the schemes were adjusted so that the final bit rates were very close to each other in order to have a fair comparison.

The parameters to control the bit rate in a normal motion-compensated scheme are the search area and the quantization for the luminance and the color components. Block classification provides four additional parameters, namely, θ_1, θ_2, ϕ_1 and ϕ_2. Of these, θ_1 and ϕ_1 are the most important, since they control the number of type *1* blocks in relation to type *2*. If θ_1 is small, the number of type *1* blocks is also small. Same is true for ϕ_1 because, in simple words, we are trying to increase the energy of type *1* blocks by increasing either the absolute value (θ_1) or the cardinality of the set $\mathcal{S}_\infty$ (ϕ_1). The result is shown in Figure 7. It is clear from the figure that

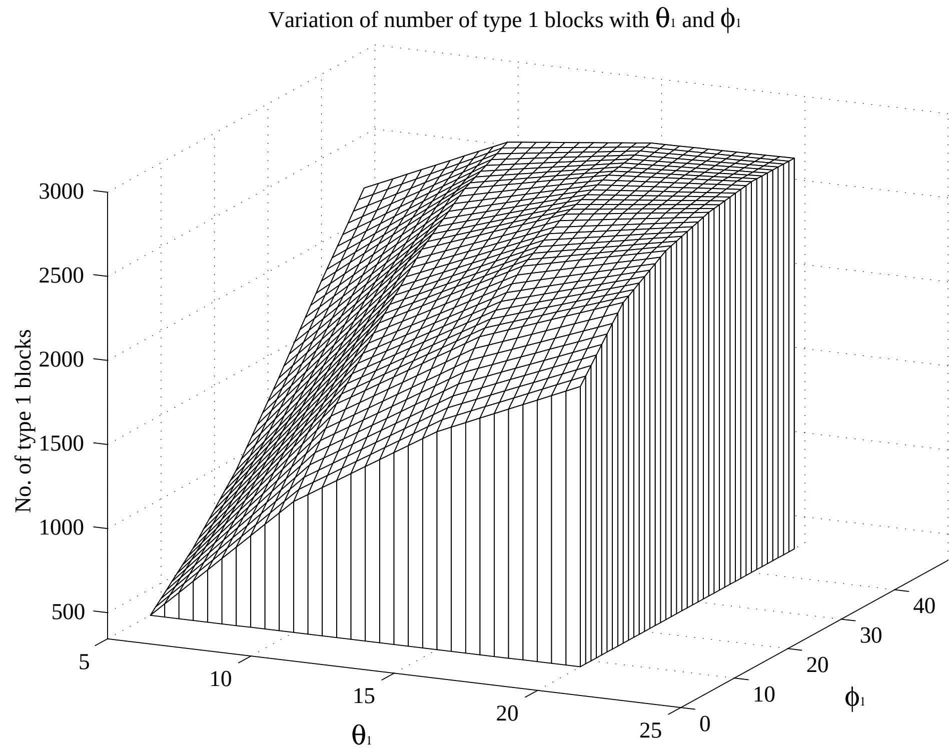


Figure 7: Variation of Number of Type *1* blocks with θ_1 and ϕ_1

increasing θ_1 by a very small value has a higher impact on the number of type *1* blocks than by increasing ϕ_1. The figure also shows that by increasing either θ_1 or ϕ_1, type *1* blocks start dominating over type *2* (which are motion compensated) and the motion-compensated video encoder asymptotically approaches to a one with simple frame difference and no motion compensation.

The variation in instantaneous bit rate for a typical frame *vs* θ_1 and ϕ_1 with all other parameters kept constant is shown in Figure 8. As expected, the bit rate flattens out for higher values of either θ_1 or ϕ_1 because at their extremes all the blocks are identified as type *1* and there is no motion compensation. The lowest achieved rate is $1.4Mbps$ and the maximum is $1.64Mbps$ for the range of parameters shown. It is clear from this figure that bit rates lower than those achieved with simple frame difference are possible with block classification and yet obtain a higher signal-to-noise ratio than that at a higher bit rate. At this lowest bit rate point, the number of type *1* blocks dominate type *2* by a factor of three to one.

The most important point to mention here is the fact that other combinations of the two parameters in question also generate the same number of type *1* blocks,

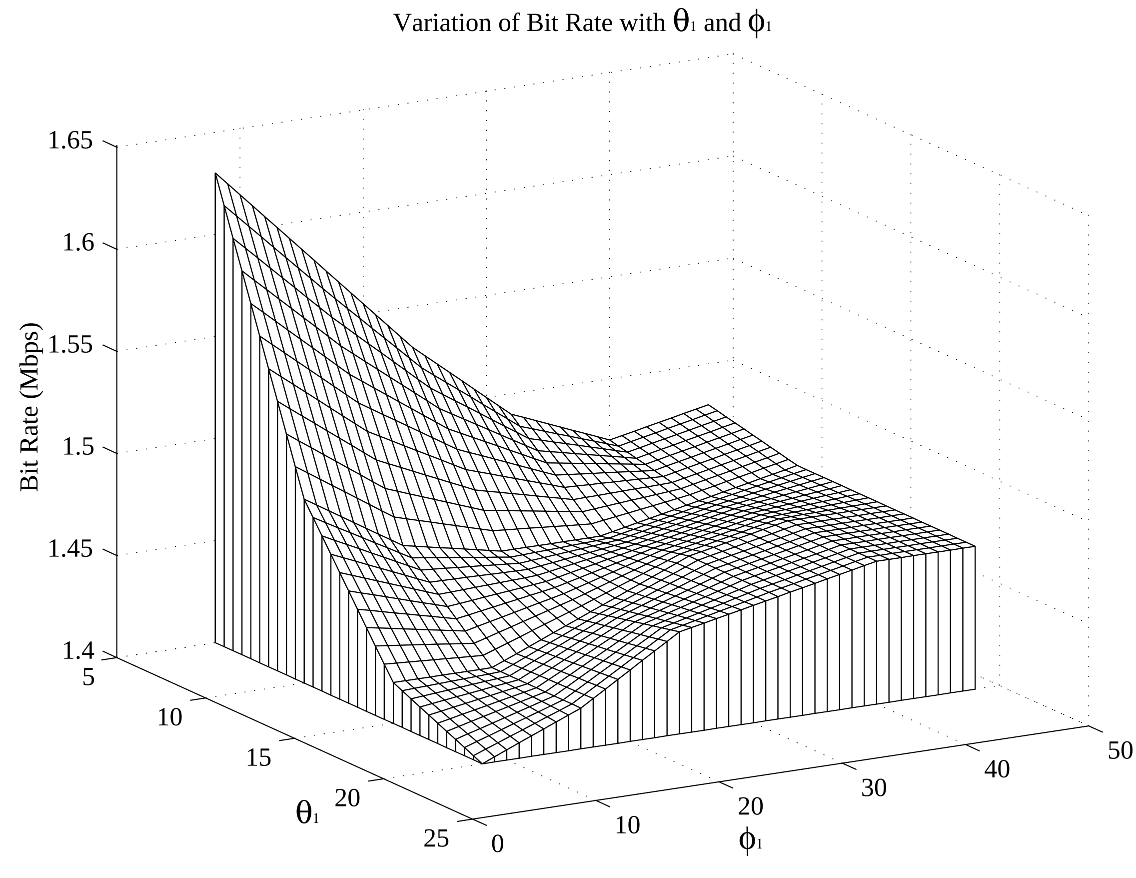


Figure 8: Variation of Output Bit Rate with θ_1 and ϕ_1

but the bit rate and signal-to-noise ratio are different in all cases. This is due to the fact that different parameters select different blocks as type *1* candidates and thus have different energies in them, though the total number of such blocks is the same. It is quite intuitive that a block with a very small number of pixels above a certain threshold will have less energy than a block with very large number of pixels above a threshold with a low value. In other words, keeping ϕ_1 low, we ensure that in type *1* blocks there will be no more than ϕ_1 pixels that have an absolute value greater than θ_1; all the rest (majority) of the pixels will be below θ_1. In the other case, a block is of type *1* if fewer than ϕ_1 pixels have a value greater than θ_1, but does not say how many of such pixels are there. Therefore, a block may be a categorized as type *1* in the first case but not in the second case and vice versa.

Figure 9 shows the profile of signal-to-noise ratio *vs* θ_1 and ϕ_1. The graph reveals that the notion of higher bit rate translating to higher signal-to-noise ratio is no longer true for all cases. Indeed, with appropriate choice of parameters (θ_1

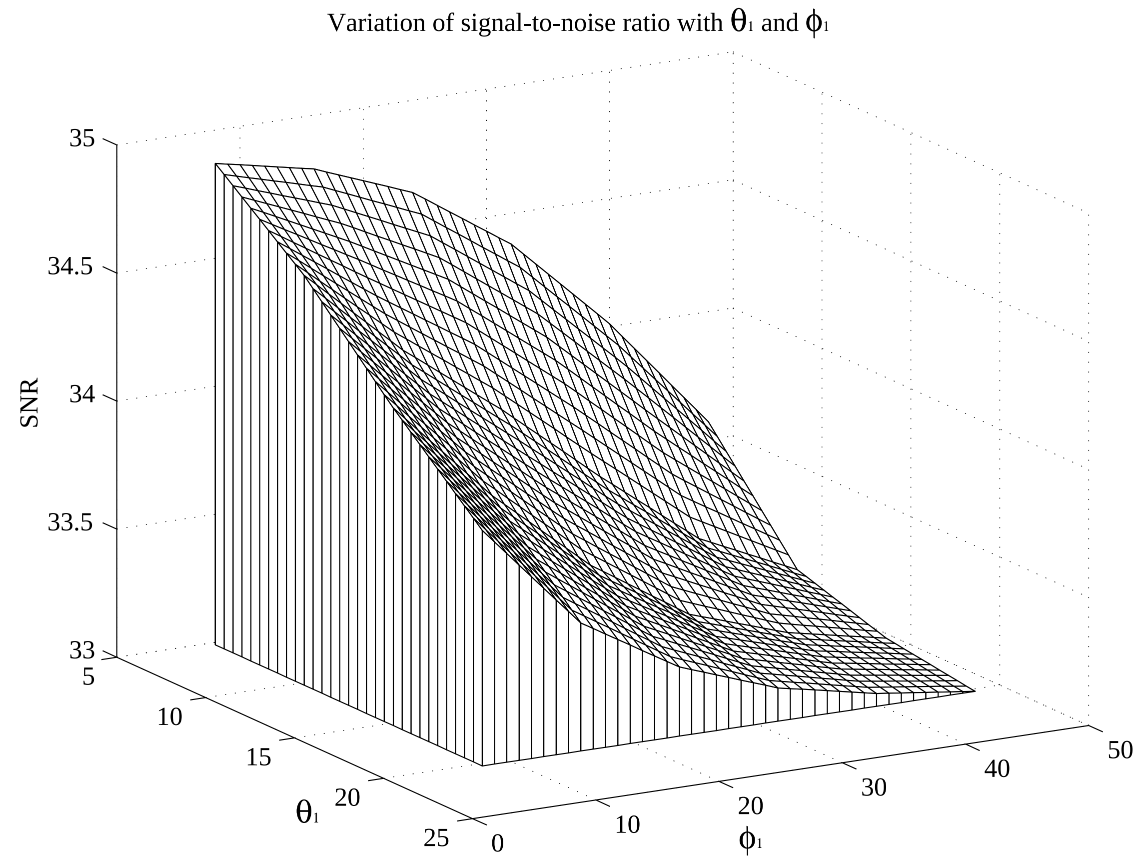
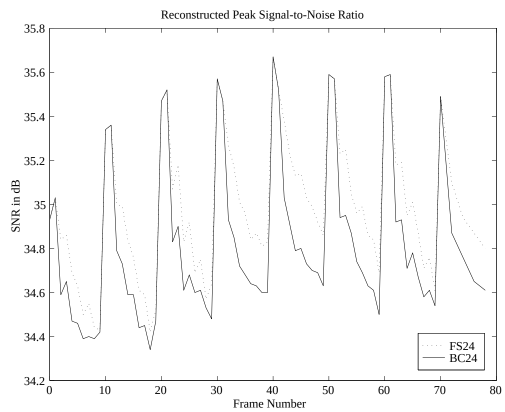


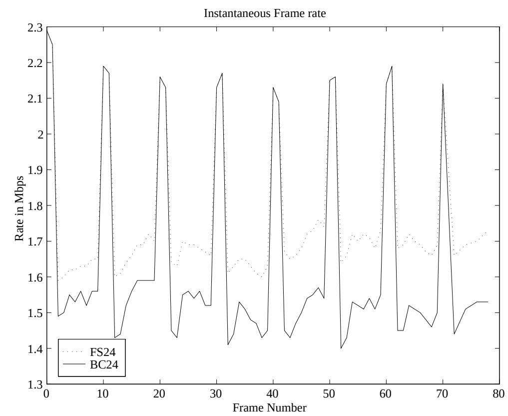
Figure 9: Variation of Signal-to-Noise Ratio with θ_1 and ϕ_1

and ϕ_1), we can improve the signal-to-noise ratio for a fixed bit rate or for a fixed signal-to-noise ratio decrease the bit rate. It should be noted that all this is achieved by a decrease in search complexity rather than an increase as one would expect. The profile complements the observations from the bit rate curve (Figure 8) that keeping a small value for ϕ_1 performs better both in terms of signal-to-noise ratio as well as bit rate. The drop in signal-to-noise ratio is small and the drop in bit rate large resulting from fixing ϕ_1 at a small value and varying θ_1 as compared with a fixing θ_1 at a low value and varying ϕ_1. This is demonstrated in Figure 10.

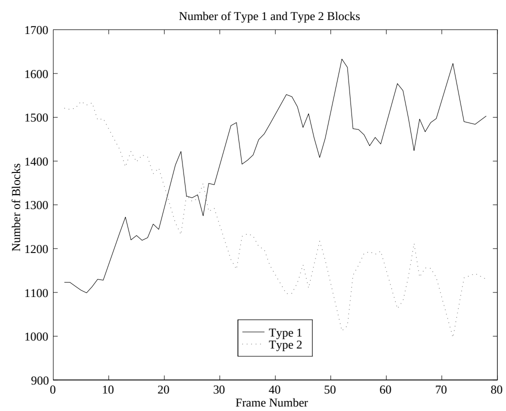
The values for $\theta_1 = 10$, $\phi_1 = 8$, $\theta_2 = 8$, and $\phi_2 = 32$ were chosen such that the reconstructed signal-to-noise ratios for the nonclassified scheme and block-classified schemes are very close to each other. Figure 10(a) shows that the difference between the two schemes is insignificant (0.2 dB). The resulting bit rate generated is shown in Figure 10(b). The difference in bit rate ranges from 0.1 Mbps to 0.3 Mbps, which is a significant improvement. Apart from the improvement in bit rate, we



(a) Signal-to-Noise Ratio



(b) Bit Rate



(c) Block Type

Figure 10: Bit Rate, Signal-to-Noise Ratio, and Block Type for $\theta_1 = 10$, $\phi_1 = 8$, $\theta_2 = 8$ and $\phi_2 = 32$.

gain significantly in the search complexity. The search complexity improvement is directly proportional to number of type *1* blocks shown in Figure 10(c) because for these blocks the motion vectors are not calculated.

Similar results hold true for θ_2 and ϕ_2, which control the number of type *3* blocks in relation to type *2* by categorizing them as type *3* if after motion compensation the energy in the block is still high. The greater the number of type *3* blocks, the greater is the bit rate, and obviously an increase (but not significant) in the signal-to-noise ratio. Type *3* blocks do not contribute to improvement in search complexity either; In fact for a type *3* block the effort to find motion vectors is wasted.

V Concluding Remarks

A novel scheme for block-based motion compensation is introduced in which a block is classified according to the energy that is directly related to the motion activity it represents. This classification allows more flexibility in controlling the bit rate and the signal-to-noise ratio and results in a reduction in motion search complexity.

The method introduced is not dependent on the particular type of motion search algorithm implemented and can thus be used with any method assuming that the underlying matching criterion used is minimum absolute difference. It has been shown that the method is superior to a simple motion compensation algorithm where all blocks are motion compensated regardless of the energy resulting after the displaced difference.

The main advantages of using such a scheme are fourfold. The first is that it allows each type of block to be treated with different priority in terms of quantization, thus improving the overall signal-to-noise ratio. Blocks that are uncompensable contain most energy as they belong to freshly uncovered regions can be quantized with more bits than those that are compensable. The blocks that do not require any compensation may be quantized with the least amount of bits, thus saving in overall bit rate. The second is that it decreases the search complexity, since no motion vectors are searched for the blocks that do not require compensation. The third is that it reduces the motion overhead, since information needs to be sent only for blocks that are motion compensable. The biggest advantage this scheme has is its ability to automatically handle abrupt scene cuts because almost all the blocks will be automatically classified uncompensable and that frame will be handled just like a refresh frame. The only disadvantage is the information of blocktype which adds to the bit rate, but this has been shown to be more than compensated by the decrease in motion overhead.

References

[1] S. Brofferio and F. Rocca, "Interframe redundancy reduction of video signals generated by translating objects," *IEEE Trans. on Communications*, vol. 25, pp. 448 – 455, April 1977.

[2] C. Cafforio and F. Rocca, "Method for measuring small displacements of TV images," *IEEE Trans. on Information Theory*, vol. 22, pp. 573 – 579, September 1976.

[3] D. L. Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, pp. 46 – 58, April 1991.

[4] B. Jabbari, F. Yegenoglu, Y. Kuo, S. Zafar, and Y. Zhang, "Statistical Characterization and Block-Based Modeling of Motion Adaptive Coded Video," *IEEE Trans. on Circuit and Systems for Video Technology*, April 1993.

[5] J. Limb and J. Murphy, "Measuring the speed of moving objects from TV Signals," *IEEE Trans. on Communications*, vol. 23, pp. 474 – 478, April 1975.

[6] F. Rocca and S. Zanoletti, "Bandwidth reduction via movement compensation on a model of the random video process," *IEEE Trans. on Communications*, vol. 20, pp. 960 – 965, October 1972.

[7] G. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM*, vol. 34, pp. 30 – 44, April 1991.

[8] F. Yegenoglu, B. Jabbari, and Y.-Q. Zhang, "Motion Classified Autoregressive Modeling for Variable Bit Rate," *IEEE Trans. on Circuit and Systems for Video Technology*, vol. 3, pp. 42 – 53, Feb. 1993.

[9] S. Zafar, *Motion Estimation and Encoding Algorithms for Hierarchical Representation of Digital Video*. PhD thesis, George Mason University, 1994.