CCITT SGXV — Doc. #376

Working Party XV/4

Specialists Group on Coding for Visual Telephony

Source : The Netherlands

Title : Improvement Reference Model 5 by a noise reduction filter

1 Summary

In the hybrid DPCM/DCT codec[1] *digital image-sequences are coded at a bitrates of* $p \times 64 kbit/s$. *There is a certain amount of distortion introduced in the form of visible artefacts. With a fixed spatial lowpass filter ([1 2 1]-filter) after the frame-memory the visual quality of the decoded images can be improved. In order to achieve a further improvement of the visual picture quality, filtering in the coding-loop and post-processing have been studied. In this paper a method is described for the improvement of the subjective picture quality.*

For the filter in the coding-loop different tasks can be distinguished;

- *noise reduction due to quantizer mismatch,*
- *improvement prediction*

Noise-reduction *needs to be carried out before the frame-memory where* prediction *needs to calculated after the frame-memory.*

With an adaptive noise-filter before the frame-memory a better visual quality can be achieved than with the [1 2 1]-filter after the frame-memory. The adaptive noise-filter consists of a concatenation of two one-dimensional sub

[1] as described in doc. 375

filters. The one-dimensional sub-filters differ in their two-dimensional orientation.
The visual picture quality with both the adaptive noise-filter and the [1 2 1]-filter in the codec is not important better than with only the adaptive noise-filter. But for the videophone application it is worthwhile to consider the implementation.

The adaptive noise-filter can also be used outside the coding loop as a post-processor in the decoder. The visual picture quality with the noise-filter outside the loop is significant better than without post-processor, altough with the noise filter in the loop better results can be achieved.

Contents

2 Introduction

Digital image-sequences which are coded at a bitrate of 64 kbit per second in the hybrid DPCM/DCT codec (Reference Model 5) contain visible artefacts due to the quantizer in the coding scheme. In this paper a noise filter is described which can be used in two ways improving the subjective quality of the decoded images. It is possible to reduce noise with the noise filter situated in the coding loop or with the noise filter as a post-processor outside the loop in the decoder. Both situations will be discussed and a comparison of the results is made.

3 The adaptive noise filter

3.1 Considerations filtering in the loop

In [1] it was decided to adopt a filter after the frame memory based on a convolution of a filter with an impulse response [1 2 1]. This filter is situated in the loop on the left side of the the frame memory, the configuration is depicted in figure 1.

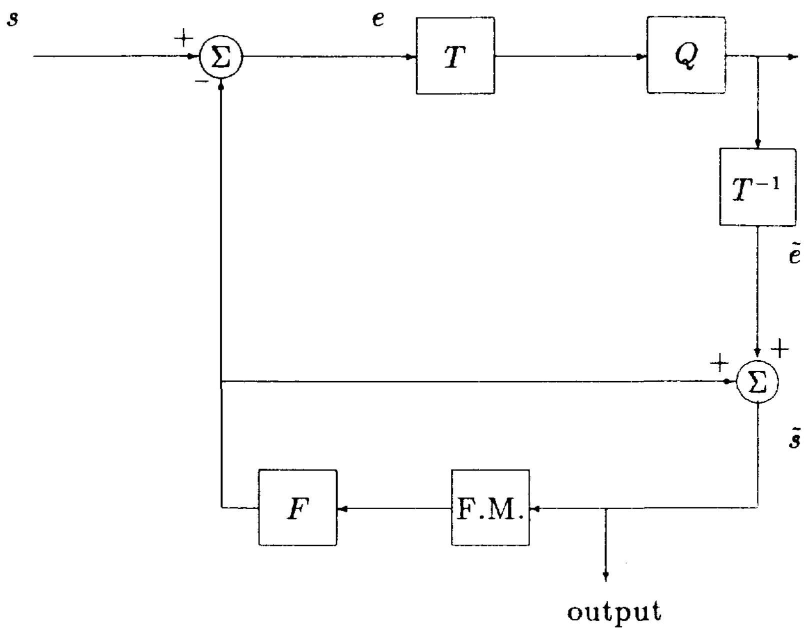


Figure 1: Position of the filter in the Reference Model

The introduction of a low pass filter after motion compensation (MC) could have the following advantages:

- A reduction of high frequency artifacts introduced by MC.
- A reduction of quantization noise in the feedback loop.

The filter could be controlled with:

- Displacement vector
- Prediction error

In order to study the quantization defects it is proposed to introduce a filter in front of the frame memory[2] as depicted in figure 2

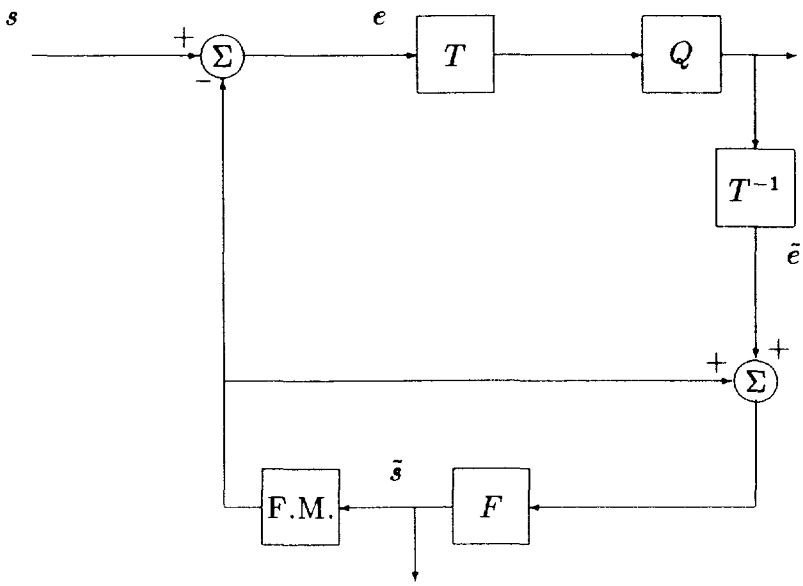


Figure 2: Filter in front of the frame memory

In [2] it was shown that the filter after the frame memory serves only one purpose i.e. the reduction of the prediction error. If motion compensation is adopted the ME mismatch[3] could be compensated. Also some sort of fractional displacement could be carried out by the character of the prediction filter.

The question arises how to compensate the quantization noise, by theoretical analysis it can be proven that one filter[4] is insufficient. We therefore introduce a second filter as depicted in figure 3 and figure 4

[2]The noise reduction filter
[3]the introduction of HF noise
[4]in the case motion compensation is adopted

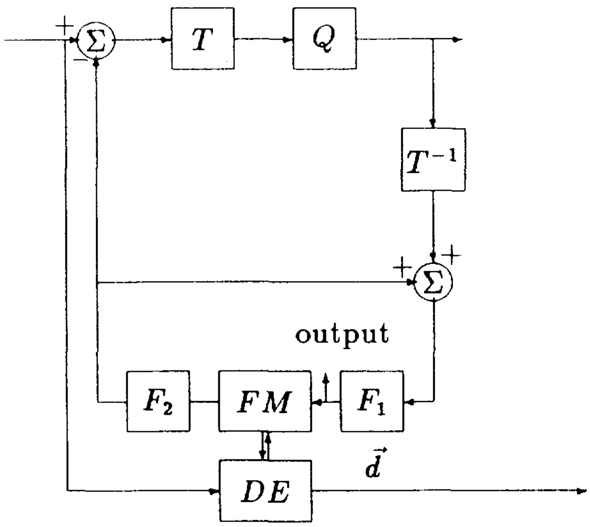


Figure 3: The encoder with noise reduction and prediction filter

3.2 The noise filter as post-processor

If the noise-filter is used outside the loop as a post-processor in the decoder, the coder and the decoder are built up as depicted in figure 5 and figure 6.

The encoder and the decoder are here the same as in RM5; although at the decoder there has to be the possibility to extract side-information for the noise-filter from the decoder.

Noise reduction by post-processing has the advantage that the encoder can be the same as described in RM5. On the other hand it is a disadvantage that the coding-noise will circulate in the loop during the coding process which is not the case with a noise filter in the loop.

4 Noise filter algorithm

The two dimensional filter is constructed by a concatenation of two one dimensional filters. The filters have an edge preserving property.

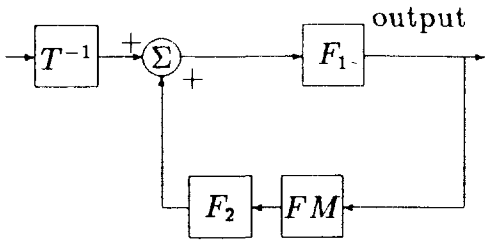


Figure 4: Decoder with prediction and noise reduction filter

Let $s(i,j)$ be the image[5] of size $dx \times dy$:

$$y(i,j) = s(i,j) + n(i,j) \tag{1}$$

The prediction denoted $\hat{s}(i,j)$ of $s(i,j)$ can be expressed with MSE as distortion measure by:

$$\hat{s}(i,j) = \mu_s(i,j) + \frac{\sigma_s^2(i,j)}{\sigma_s^2(i,j) + \sigma_n^2(i,j)} \cdot \{y(i,j) - \mu_y(i,j)\} \tag{2}$$

Denote $\mu_s(i,j)$ the estimate of the average of the noise free signal, $\mu_y(i,j)$ the estimate of the noisy signal, $\sigma_s^2(i,j)$ the variance of the signal and $\sigma_n^2(i,j)$ the variance of the noise.
The estimate for the expectation $\mu_y(i,j)$ and the variance $\sigma_y^2(i,j)$ of y in an certain position (i,j) can be derived with the local variance and local average in a so called observation area. This observation area is defined around the position (i,j) for which the statistics need to be calculated.

Let the window for the observation area be a of size $2n+1$. Now the 1-dimensional filter [6] can be expressed with [3] [4]:

[5] with additive noise $n(i,j)$ where the average is 0 and un-correlated with the source signal

[6] suppose for horizontal direction first

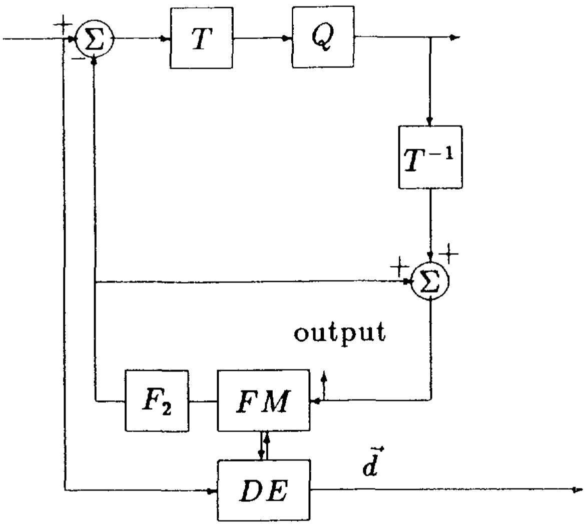


Figure 5: The encoder with prediction filter

$$\hat{s}_1(i,j) = \hat{\mu}_s(i,j) + \frac{\hat{\sigma}_s^2(i,j)}{\hat{\sigma}_s^2(i,j) + \sigma_n^2(i,j)} . \{y(i,j) - \hat{\mu}_y(i,j)\} \tag{3}$$

with:

$$\hat{\mu}_s(i,j) = \hat{\mu}_y(i,j) = \frac{1}{2n+1} \sum_{k=i-n}^{i+n} y(k,j) \tag{4}$$

Suppose that signal and noise are uncorrelated the expression can be formulated with:

$$\hat{\sigma}_s^2(i,j) = \begin{cases} \hat{\sigma}_y^2(i,j) - \sigma_n^2(i,j) & \text{if } \hat{\sigma}_y^2(i,j) > \sigma_n^2(i,j) \\ 0 & \text{else} \end{cases} \tag{5}$$

$$\hat{\sigma}_y^2(i,j) = \frac{1}{2n} \sum_{k=i-n}^{i+n} \{y(k,j) - \hat{\mu}_y(i,j)\}^2 \tag{6}$$

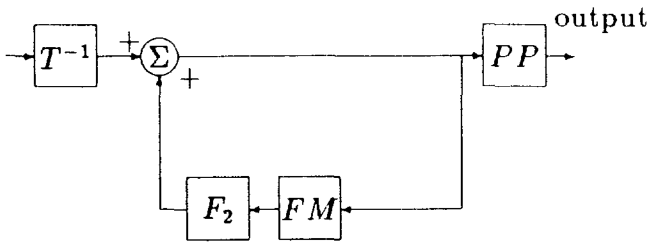


Figure 6: The decoder with prediction filter and post-processing

The resulting noise filter consists of a concatenation of N one dimensional sub-filters. In the case $N = 4$ the expression becomes:

$$\hat{s}(i,j) = T_4[T_3[T_2[T_1[y(i,j)]]]] \tag{7}$$

5 Implementation

5.1 Filter in the loop

The noise filter consists of sub-filters. Each filter can have a different spatial orientation. Four spatial orientations will be considered. The sub-filter orientations are depicted in figure 7.

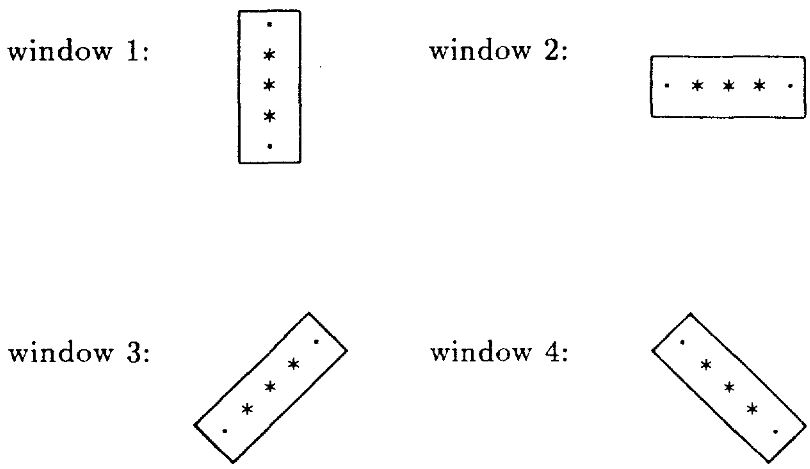


Figure 7: Example 1-dimensional windows

In this figure the spatial orientation is called a window. A window consists of five pixels on a line. The noise filter operates over the block boundaries of an 8×8 pixel input block. Figure 8. shows the blocks which need to be accessed by the filter when the input block X is being filtered.

The block which are needed for this filter are denoted by A, B, C and D. It depends on the window of a sub-filter which adjacent blocks are needed for the sub-filter operations.

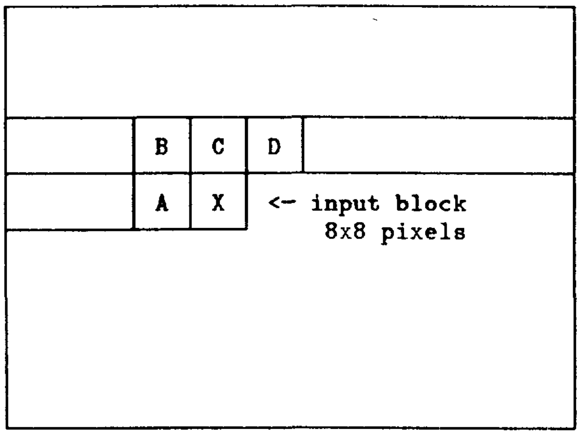


Figure 8: Memory access

If window 1 is used then block C is needed.
If window 2 is used then block A is needed.
If window 3 is used then the blocks A, C and D are needed.
If window 4 is used then the blocks A, B and C are needed.

For the pixels on the bottom and the right-hand side of the input block X the adjacent blocks are not available at the time of processing the input block X. In this case the pixels of the input block are copied. To implement the noise filter a buffer memory is needed which has a size of a least one line of blocks. This memory could be a part of the frame memory.

The noise filter operates on luminance and chrominance blocks of 8x8 pixels. During the sub-filter operations the following steps can be distinguished:

- Calculation of the average of the window pixels.
- Calculation of the variance of the window pixels.

- Filtering of the pixel.

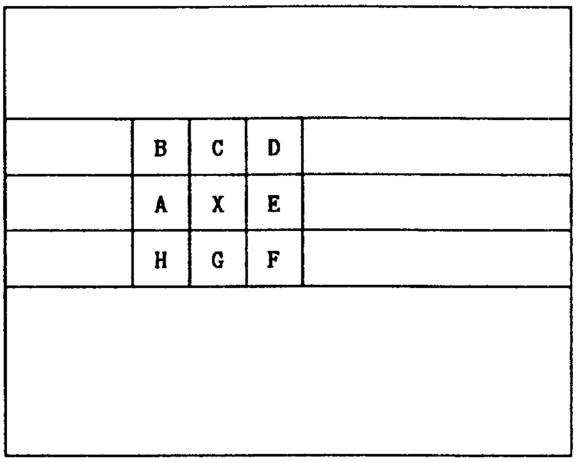
For the calculation of the variance of the window pixels the noise variance is needed. This value can be calculated in the encoder by using the source input data and the reconstructed data. Because the source input data is not available in the decoder this calculation is not possible in the decoder. A way to solve this problem is by using estimated noise variance values. Estimated values of the noise variance can be stored in tables in the decoder and in the encoder. These values can be conditional on the quantiser step size and the motion of the block. The estimated noise variance can be found by calculations applied to a number of test sequences. Using these estimated values will not give an optimal performance but simulations have shown the this method gives good results.

Two different tables are needed i.e. one for the luminance and one for the chrominance blocks.

5.2 Filter outside the loop

Comparing the case of noise removal during post-processing to noise filtering in the loop there are two differences.

First, it is possible now to acces all the adjacent blocks which are needed to filter the current block. The blocks which are needed here are denoted by A,B,C,D,E,F,G and H. It depends on the window of the sub-filter which adjacent blocks are needed.



X: filtered block
8x8 pixels

If window 1 is used then block C and G are needed.
If window 2 is used then block A and E are needed.
If window 3 is used then the blocks A, C, D, E, G and H are needed.
If window 4 is used then the blocks A, B, C, E, G and F are needed.

Because all adjacent blocks are available it is now not necessary to copy the pixels of the input block at the right-hand side and the bottom. This makes it possible to affect the blocking-effect efficiently.

Second, the input of the noise filter differs here from the input of the noise filter in the case that the filter is situated in the loop. When there is no explicit noise reduction in the coding loop, the noise can cause a higher stepsize of the quantizer.

The post-processor operates on luminance and chrominance blocks of 8x8 pixels. For the estimation of the noise variance two tables (one for the luminance and one for the chrominance) are used in the decoder. The chosen entry in the table depends on the quantizer stepsize and the motion-vector of a block. To implement the noise filter a buffer memory is needed which has a size of a least one line of blocks.

6 Conclusion

- Simulation results have shown that the introduction of a noise filter in front of the frame memory and a noise filter as post-processor both will give a significant improvement of the subjective picture quality of a codec working at 64 kbit/s.

- With the noise filter in the loop a better subjective picture quality can be achieved (i.e. a more efficient surpression of the mosquito-effect) than with the noise filter outside the loop as post-processor.

References

[1] CCITT SGXV/1 Working Party 4 Specialists Group on Coding for Visual Telephony, Description of Reference Model 5, Document 339, 1988, version march 17

[2] The efficiency of motion-compensating prediction for hybrid coding of video sequences. B. Girod. IEEE Journal on selected areas in communications. vol. SAC-5 no. 7. August 1987 pp. 1140-1154.

[3] One-dimensional processing for adaptive image restoration. P Chan, J. S. Lim. IEEE International conference on acoustics, speech and signal processing. San Diego, California. March 19-21 1984, pp. 37.3.1-37.3.4

[4] Digital image enhancement and noise filtering by use of local statistics. Jong-sen Lee. IEEE Transactions on pattern analysis and machine intelligence. vol. PAMI-2, no. 2, March 1980, pp. 165-168.