HUITIEME COLLOQUE SUR LE TRAITEMENT DU
SIGNAL ET SES APPLICATIONS

NICE du 1er au 5 JUIN 1981

VIDEO HYBRID CODING BY PICTURE DOMAIN SEGMENTATION

Osvaldo Telese Giovanni Zarone

Istituto Elettrotecnico, Facoltà d'Ingegneria, Università degli Studi di Napoli
21 via Claudio, 80125 Napoli, Italy

RESUME

La codification hybride d'images s'avère une technique très efficace, car elle exploite à la fois la corrélation spatiale (grâce à la transformation à deux dimensions "intraframe") et la corrélation temporelle (grâce au DPCM "interframe"), sans nécessiter la capacité de mémoire demandée par une transformation à trois dimensions.

Le schéma qu'on analyse ci-après a recours à une segmentation dans le domaine de l'image (plutôt que dans le domaine de la transformée) et à l'agrégation des éléments mis en rectangles. En conséquence on peut se limiter à transformer seulement le signal différence de cadre relatif à ces blocs, au lieu de l'image entière, et les informations d'adresse nécessaires en résultent réduites.

La qualité d'une séquence vidéo-téléphonique, traitée selon cette codification et utilisant les transformées de Fourier, Hadamard et Coseno, avec et sans filtrage, a étée évaluée par des paramètres objectifs, au moins de la représentation sur un moniteur et, enfin, au moins d'un paramètre se rattachant directement aux opinions subjectives des observateurs.

SUMMARY

The hybrid transform/predictive coding of correlated frames is an efficient technique because it exploits both the spatial correlation (through the 2-D intra frame transform) and the temporal correlation (through interframe DPCM) without the storage requirement of a 3-D transform.

The investigated adaptive scheme adopts a picture domain segmentation (instead of a transform domain one) and the aggregation of moving pels in rectangles. This allows to transform just the frame difference signal pertinent to these blocks, rather than the whole images, and reduces the necessary address information.

The quality of a videotelephonic sequence, processed by this coding and using Fourier, Hadamard and Cosine Transforms, with and without any filtering, was evaluated by objective parameters, by displaying it on a monitor and by a parameter tied to subjective opinions of observers.

VIDEO HYBRID CODING BY PICTURE DOMAIN SEGMENTATION

1 - Introduction

The hybrid transform/predictive coding [1, 2] is an effective technique to reduce the bit rate of video telephonic signals. In fact it retains some attractive features of both the transform and the DPCM techniques. An interframe hybrid coder partitiones each video frame into blocks of smaller size and transforms each block into a set of coefficients. Each one of the transform coefficients is linear predicted in an interframe DPCM loop using the coefficients from the previously transmitted and temporally adjacent block as a prediction. The sequence of quantized prediction errors is then transmitted to the receiver; here the inverse operations, DPCM decoding and inverse transform, are performed.

The hybrid coder exploits both the spatial and the temporal correlation between pels, through the intraframe transform and the interframe DPCM respectively. Nevertheless the storage requirement is little compared with systems which use 3-D transform: one frame memory is necessary at the transmitter and at the receiver. Moreover, for scenes with low detail and small motion, many coefficient-differences have little energy; they can be either dropped or coarsely quantized without effecting the picture quality appreciably.

This coder can profit to a great extent by the conditional replenishment technique, which performs a segmentation of each frame into changed and unchanged parts with respect to the reference frame. Only information about the changed part is transmitted; it updates both the reference memory of the coder and the frame memory of the decoder. Of course, addressing information is necessary. However a considerable reduction in transmission costs is expected by employing this technique, which usually is named "conditional replenishment in the transform domain".

Fig. 1 shows a scheme of a possible implementation of the conditional replenishment transform coder [3]. The figure is self-explanatory. We note that the coder in fig. 1 employs an interesting segmentation in the transform domain. This domain is divided in two regions:

- a predictable region, in which the receiver's estimate is regarded as adequate;
- a non predictable region, in which the receiver's estimate is not adequate and updating information need to be transmitted.

Such a segmentation has a serious drawback. Generally each block contains both changed and unchanged pels with respect to the reference block. The energy of significant frame-differences is scattered all over the block after the transform operation. Therefore, significant frame-differences, located in small areas of the block, can give rise to many non significant differences between homologous spectral components; if such differences are below the threshold for predictability, they are irreversibly lost. The drawback affects more the large size blocks, which, on the other hand, are those allowing a more effective redundancy removal. Moreover, it is not very efficient to transform and to process each block, even if it belongs to the fixed area.

The above reasons induced us to approach the hybrid coding problem by performing the segmentation in the image domain. This kind of segmentation has been widely tried and efficient algorithms for subdividing a frame into fixed and moving areas are at present available in the technical literature [4, 5].

2 - The investigated conditional replenishment transform scheme

Our conditional replenishment transform coder (fig. 2) involves three functional blocks: an image processor, a frame-difference processor and a code assigner.

The image processor recognizes in the present frame single bounded regions, including for the most part contigous moving pels; then it computes the address information about these regions and sends it either to the frame-difference processor or to the code assigner. To be explicit, the image processor performs a segmentation and an "aggregation". The segmenter detects the pels which are moving. Its input is the frame-difference signal; its output is a binary information denoting whether the processed pel is moving or fixed. The "aggregator" processes the information about the location of moving pels in order to recognize rectangular blocks, as large as possible,provided that they include, for the most part, moving pels.

Once a "moving" block has been located, its address information is sent to the frame difference processor; only the frame differences included in the "moving" blocks are processed further, through a block-transformation and a quantization of the transform coefficients. In the feedback chain, an inverse transformation is required in order to make a prediction of the sample to be encoded from previously transmitted information.

Finally the code assigner represents the data by means of codewords of variable length and processes the address information about the rectangular blocks. A time-multiplexed address and data signal is then transmitted to the receiver.

Our scheme presents the following positive features:
1) The segmentation is not subsequent to the transform operation,so the reconstruction of significant frame differences is not impaired and large blocks may be used in order to achieve an effective redundancy removal.
2) It is enough to transform just those areas which in=

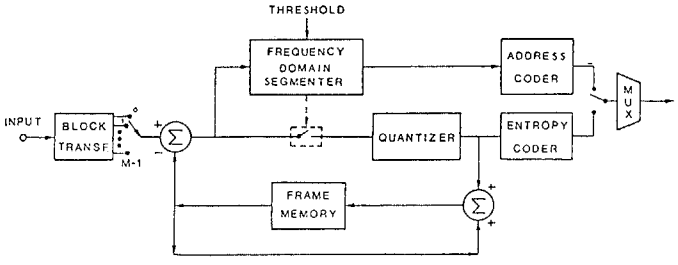

Fig. 1 - Hybrid coder by transform domain segmentation

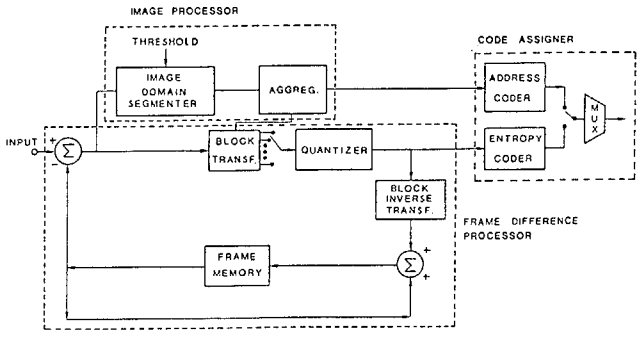

Fig. 2 - Hybrid coder by picture domain segmentation

clude moving pels for the most part, not the whole image. So, if the scene has a small content of movement, the required processing is limited.
3) The partition of the picture into blocks is not fixed once and for all, but it matches the signal evolution.
4) The address information is cheaper than that required by the coder in fig. 1: it is necessary to transmit only the locations of the "moving blocks", not the addresses of single coefficients.

The identification of rectangular blocks, including for the most part moving pels, can be easily implemented by a fast algorithm.

3 - The aggregation

The aim of the aggregation algorithm is to identify, inside an image, rectangular "moving" blocks whose dimensions are as large as possible compatibly with the condition that the ratio between the number of moving pels (previously located by the segmenter) and the total number of pels included in the block is greater than or equal to a prefixed threshold T. In our simulation we chose T=0.75 and decided to limit ourselves to identifying the rectangular moving blocks whose dimensions are (m, bm), where m is an integer power of two and b=1 or 2.

The input data to the aggregation algorithm is a matrix N_o provided by the segmenter; its elements carry the information about the state (fixed or moving) of each pel in the frame:

$N_o(i,j) = 0$ if (i,j) is a fixed pel;

$N_o(i,j) = 1$ if (i,j) is a moving pel.

The output data from the aggregation algorithm is a set of M matrices. The generic matrix N_k is the map of the moving blocks whose dimensions are $(2^{\mu}, 2^{\lambda+1})$, where:

$$\mu = INT(0.5(k+1)) \qquad k = 1, 2 \ldots M$$
$$\lambda = INT(0.5k)$$
$$M = \log_2(RN.CN)-1$$

(RN, CN) are the processed frame dimensions.
As output data from the aggregation algorithm we have:

$N_k(i,j) = 0$ if the (i,j) block is fixed or included in a larger moving block;

$N_k(i,j) = 1$ if the (i,j) block is moving and not included in a larger moving block.

Our algorithm (fig. 3) consists of two subsequent sections.

First section. The construction of the matrices N_1, $N_2 \ldots N_M$ is performed in the given order and in a temporary form. The employed strategy consists of progressively compressing the scanning process of the frame in order to enucleate the largest blocks from the smallest ones. The compression of the scanning process is accomplished step by step by aggregating adjacent blocks of given size two by two and blending into the larger blocks the information about the number of moving pels. In other words, the compression is easily achieved by storing, in each element of N_k, the sum of the numbers of moving pels included in the blocks represented by 2 adjacent elements of the matrix N_{k-1}. Moreover, once an element of the matrix N_k has been constructed, the algorithm carries out a comparison with a given threshold in order to store the information about the state of the corresponding block (moving or fixed).

Second section. The definitive construction of the

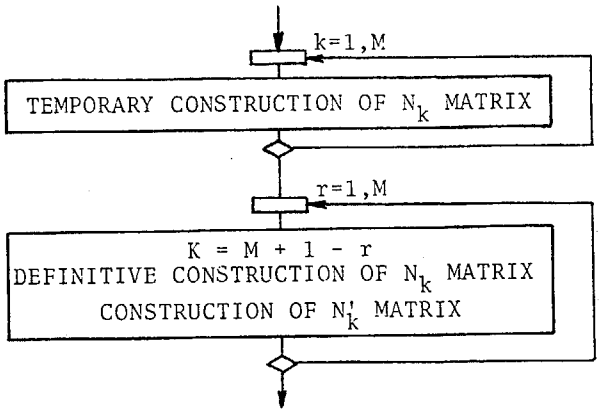

Fig. 3 - The aggregation algorithm

maps is carried out in the opposite order to the former one. Moving blocks included in greater blocks, already classified as moving, are excluded from their map: during the definitive construction of the matrix N_k, an updating of the matrices N_{k-1}, $N_{k-2} \ldots N_1$ is carried out in order to exclude those blocks which, actually, are sub-blocks of moving rectangles already mapped.

Immediately after the construction of N_k, the matrix N'_k is formed in order to reduce the rigidity by which the block positions are settled. The matrix N'_k takes into account the presence of $(2^{\mu}, 2^{\lambda+1})$ blocks which are staggered with respect to those having N_k as map.

Of course, this algorithm is just one of the possible algorithms; moreover some choices, such as the value assigned to the threshold T, could be optimized. As T increases, in fact, the percentage of fixed pels, included in the "moving" rectangles, decreases (this is a positive feature); on the other hand the trend to construct large blocks is hampered and the addressing information becomes more expensive. The set up of this algorithm just pursued the aim to show that the association of moving pels in rectangles can reached rapidly, by simple test and sum operations.

Our algorithm was used to process the test signal, a videotelephonic sequence of 16 images (Judy); every picture had 64 x 64 pels uniformly quantized by 256 quantic ranges (See [6] to know statistics regarding this signal). The ratio between the number of moving pels and the total number of pels included in the "moving" rectangles was 0.85. The efficiency of the algorithm can be deduced by the table I, where P(N) is the percentage of pels included in blocks having size N. From this table, in fact, we get that the 55% of the aggregated pels belongs to blocks having a size greater than or equal to 128 pels.

N	2	4	8	16	32	64	128	256	512	1024
P(N)	2.0	7.8	8.3	7.5	9.3	7.8	11.3	14.8	18.7	12.5

Table I

4 - Quantization of the spectral components

Once the moving blocks have been located, they are transformed into sets of coefficients. Blocks of different sizes are processed in the same frame.

The dc component of each block was quantized by an uniform and N-independent step.

The ac components were uniformly quantized and Huffmann coded; the quantization step was taken according

the following rules:

a) $q = k_a/\sqrt{N}$

b) $q = k_b\{ 1+u(\sigma-\sigma')\}/\sqrt{N}$

c) $q = k_c \sigma_q$

d) $q = k_d \sigma_q/\sqrt{N}$

where u(.) is the unit step function and σ_q is the quantized value of the standard deviation σ of the frame difference signal pertinent to the transformed rectangle. To assume

$$q \propto N^{-\frac{1}{2}}$$

makes independent on N the maximum error which affects the signal (after the inverse transformation), the other parameters being equal. The dependency of the quantization step on the standard deviation adapts the coding to the local property of the signal. In those image areas where the movement content (i. e. σ) is greater, the quantization step is in fact less fine. This behaviour matches a well known psychophisical property of the human observer, according to which the sensibility to errors decreases as the movement in the scene increases. Of course, this adaptivity requires the transmission of σ_q ("c" and "d" cases) or the transmission of 1 bit/rectangle ("b" case) in order to inform the receiver whether σ is lower or higher than a prefixed threshold σ'.

5 - Filtering

Once the block tranform of the frame difference signal is performed, it is possible to carry out a simple filtering by cutting off the more peripherical components of the bidimensional spectrum. Therefore, not only the coder shown in fig. 2 was computer simulated, but a system which carries out the cutting off of some spectral components (between the block transformer and the quantizer) was simulated too. The remaining components were quantized according the d law of the previous section and Huffmann coded.

6 - Performance of the investigated coding scheme

The coding scheme of fig. 2 was computer simulated using Judy as the test signal.

The objective impairement was evaluated by means of the r. m. s. error $g = \sqrt{\overline{e^2}}$ for the various sized block. The Cosine transform proved slightly better than Hadamard and Fourier Transforms for the greatest blocks. This supremacy disappeared for the smallest blocks. In fact the cost, in bits/pel, for transmitting the dc component and the standard deviation takes an increasing weight as the block size decreases. Given the type of transform, the objective impairements proved depend more on the bit rate than on the adopted quantization law. However, as the size of blocks changed, the d law provided more uniform performance respect to the other laws, i. e. it made the quantization noise nearly uniform all over the picture.

The average distortion obtained all over the sequence for the three transforms, with and without any filtering, is plotted against the bit rate in fig. 4. The cutting off noise due to the filtering is more marked adopting the Hadamard Transform, since the spreading of energy in the transform domain is greater in this case. Moreover, as the bit rate increases, the performance of the scheme provided with filtering did not improve appreciably. In fact, once the quantization noise became neglectable with respect to the cutting off noise, every further reduction of the first one causes unconsiderable variation to the sum of the two contributions. After all that, the investigated coding scheme, tested by a videotelephonic sequence, seems to provide a satisfactory objective quality at 1.1 bits/pel with a normalized mean square error equal to 0.03% (1).

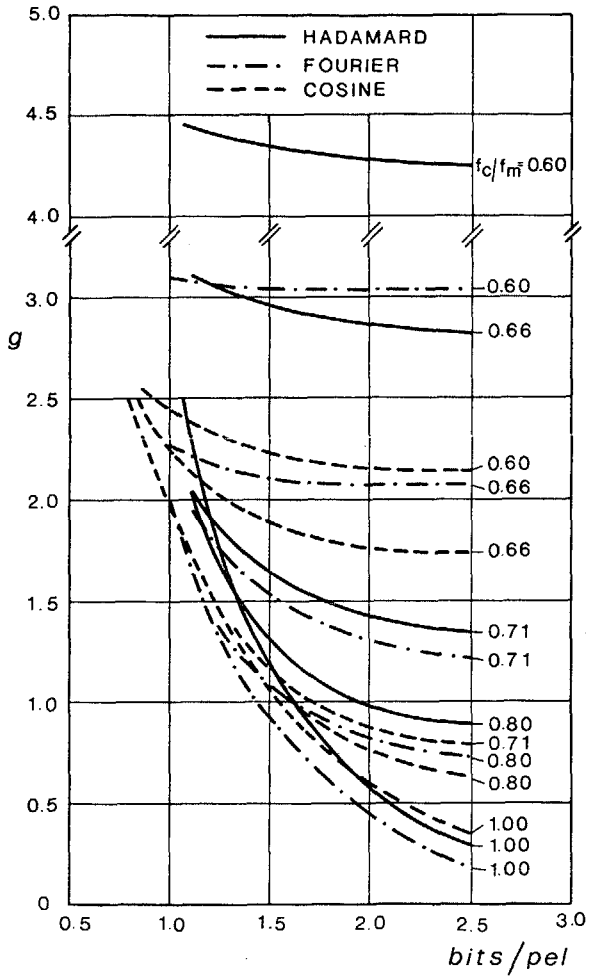

Fig. 4 - Average distorsion all over the whole sequence. f_c is the cut off frequency; f_m is the maximum spatial frequency before filtering.

In order to evaluate the subjective quality of the processed images, we displayed them on a monitor. The original 8-th image of the sequence and the corresponding processed images at 1.1 bits/pel are given in fig. 5.

Sharma and Netravali |8| introduced a distortion measure closely related to the subjective evaluation of the picture quality, the Mean Square Supra Threshold Error:

$$MSSTE = \sum_{i=1}^{L} \int_{x_{i-1}}^{x_i} \{ (x-y_i)^2 - t^2(x)\}u\{|x-y_i|-t(x)\}w(x)dx$$

where L is the number of the quantizer levels;
x is the slope of the original signal;
y is the slope of the reconstructed signal;

(1) At 1.1 bits/pel, in fact, the rms error is 1.72 (see fig. 4), while the rms value of the signal is 99.8 (see |6|).

w(x) is the "visibility function" of the quantization noise (2);

t(x) is the "visual threshold function", which measures the amount of distortion which can be added to the signal without noticeable subjective impairements.(2).

u(.) is the unit step function.

We used the MSSTE in order to evaluate the quality of various processed images. The caption of every photo gives the corresponding value of the MSSTE parameter. Actually the MSSTE was originally formulated for still images; we expect that, dealing with moving images, the resulting subjective quality would not be worse than that predictable on the ground of the asserted relationship between the MSSTE and the subjective judge of the observers for still images.

7 - Address coding

The addressing information has an important role in hybrid coding since it takes up a large fraction of the total transmitted bits. For the conditional replenishment transform coder, computer simulations on sequences of moving images |3| showed that the address information accounts for about the 50% of the total transmitted bits; no appreciable bit rate reduction was achieved by employing various addressing techniques.

Our scheme has some potential advantages compared with the conventional hybrid coders. Transmitted elements are coalesced in rectangular clusters, so the receivers knows a priori the shape of blocks identified as moving at the transmitter end. It follows that, in order to inform the receiving end about the location of a moving rectangle of given size, we have to send just one address for the whole rectangle (say the address of the uppermost and the leftermost element of the rectangle). Moreover, this absolute addressing can profit to a large extent by the fact that, inside the image, the locations of moving rectangles are not all allowed. Finally, we point out that the map of the moving blocks is directly obtained from the aggregation algorithm as output data.

An "ad hoc" addressing-coding law could be implemented in order to match the features of our coder. However, rather than implementing a new addressing technique, we just limited ourselves to use some known algorithms and to verify their effects on the total bit rate. We were aware that absolute addressing of each moving rectangle is an attractive technique for hardware implementation. On the other hand its efficiency is limited to the largest rectangles; for the smallest ones, techniques which are usual in the facsimile signal coding (run length and block coding) could result in better performance.

The above reasons induced us to test a hybrid addressing technique: absolute addressing for the largest blocks (taking into account that the positions of these blocks in the image are not all allowed); run length or block coding for the remaining (smallest) blocks.

Results obtained by the computer simulation of the proposed hybrid addressing technique are shown in table II. In this table N_T is the minimum size (in number of pels) of the blocks which absolute addressing has been applied to; P_L is the percentage of the elements which are included in the absolute addressed blocks and C_L the corresponding addressing cost in bits/pel. $P_s = 1-P_L$ is the percentage of the remaining elements and C_s the corresponding addressing cost. In our simulation we assigned three values to N_T (i. e. 16, 32, 64) and evaluated the global performance in terms of the average addressing cost $C = P_L C_L + P_s C_s$. From table II we get that the best performance ($C = 0.29$ bits/pel) is obtained with $N_T = 16$ and a 2x4 block coding.

N_T		pels	16	32	64
P_L		%	42.6	38.7	33.6
C_L		bits/pel	0.11	0.05	0.03
P_s		%	57.4	61.3	66.4
C_s	run length	bits/pel	0.51	0.52	0.51
	2x4 block coding	bits/pel	0.43	0.46	0.47
	4x2 block coding	bits/pel	0.49	0.52	0.53
C	run length	bits/pel	0.34	0.34	0.35
	2x4 block coding	bits/pel	0.29	0.30	0.32
	4x2 block coding	bits/pel	0.33	0.34	0.36

Table II

8 - Conclusions

A hybrid coding, which adopts a segmentation in the image domain, the aggregation of moving pels in rectangular blocks and the transformation of the only pels included in these blocks, was computer simulated. This coding was applied to a test signal made up by a sequence of videotelephonic pictures.

The performance of this scheme for some transforms (Fourier, Cosine and Walsh-Hadamard transforms), with and without any filtering, were determined. The address coding was investigated too.

The quality of the processed images was evaluated by objective parameters, by displaying them on a monitor and by a parameter (MSSTE) strictly tied to the subjective opinions of the observers.

Although many parameters of the coding scheme were not optimized yet, the results show the possibility to achieve satisfactory performance.

A further study could be devoted to the optimization of the aggregation algorithm, to the use of other transforms and to more sophisticated filtering techniques.

The authors wish to thank Prof. F. Rocca for helpful discussions and Prof. C. Cafforio who allowed to display the processed images.

(2) In order to determine the visibility function w(x), we followed an approach due to Limb |8| who expressed it as:

$$w(x) = p^{\alpha}(x)/m(x)$$

thus stressing the effect of two components on the visibilily of the quantization noise:

- a picture dependent one, namely the probability density function p(x), which can be easily estimated for each image being processed;
- a viewer dependent component, namely the masking function m(x); as a result of psychovisual experiments, Limb found that m(x) is closely approximated by a simple exponential:

$$m(x) = m_o \exp(-m_1 x).$$

The values of α, m_o and m_1 were taken from the table II of |8|. The visual threshold function t(x) was taken from |7|.

References

[1] A. Habibi, "Hybrid Coding of Pictorial Data", *IEEE Trans. on Comm.*, vol. COM-22, pp. 614-624, 1974

[2] J. A. Roese, W. K. Pratt, G.S. Robinson, "Interframe Cosine Transform Image Coding", *IEEE Trans. on Comm.*, vol. COM-25, pp. 1329-1339, 1977

[3] J. A. Stuller, A. N. Netravali, "Transform Domain Motion Estimation", *BSTJ*, pp. 1673-1702, 1979
A. N. Netravali, J. A. Stuller, "Motion Compensated Transform Coding", *BSTJ*, pp. 1703-1717, 1979

[4] A. N. Netravali, J. D. Robbins, "Motion Compensated Television Coding", *BSTJ*, pp. 631-669, 1979

[5] C. Cafforio, F. Rocca, "Methods for Measuring Small Displacements of Television Images", *IEEE Trans. on Inf. Th.*, vol. IT-22, pp. 573-579, 1976

[6] L. Arena, G. Zarone, "3-D Filtering of Television Signal", *Alta Frequenza*, vol. 46, pp. 108-116, 1977

[7] D. Sharma, A. N. Netravali, "Design of Quantizers for DPCM Coding of Picture Signals", *IEEE Trans. on Comm.*, vol. COM-25, pp.1267-1274, 1977

[8] J. Limb and C. Rubinstein, "On the Design of Quantizers for DPCM Coders: a Functional Relationship between visibility, probability and masking", *IEEE Trans. on Comm.*, vol. COM-26, pp. 573-578, 1978
C. Rubinstein, J. Limb, "On the Design of Quantizers for DPCM Coders: Influence of the subjective Testing Methodoloy", *IEEE Trans. on Comm.*, vol. COM-26, pp. 565-572, 1978

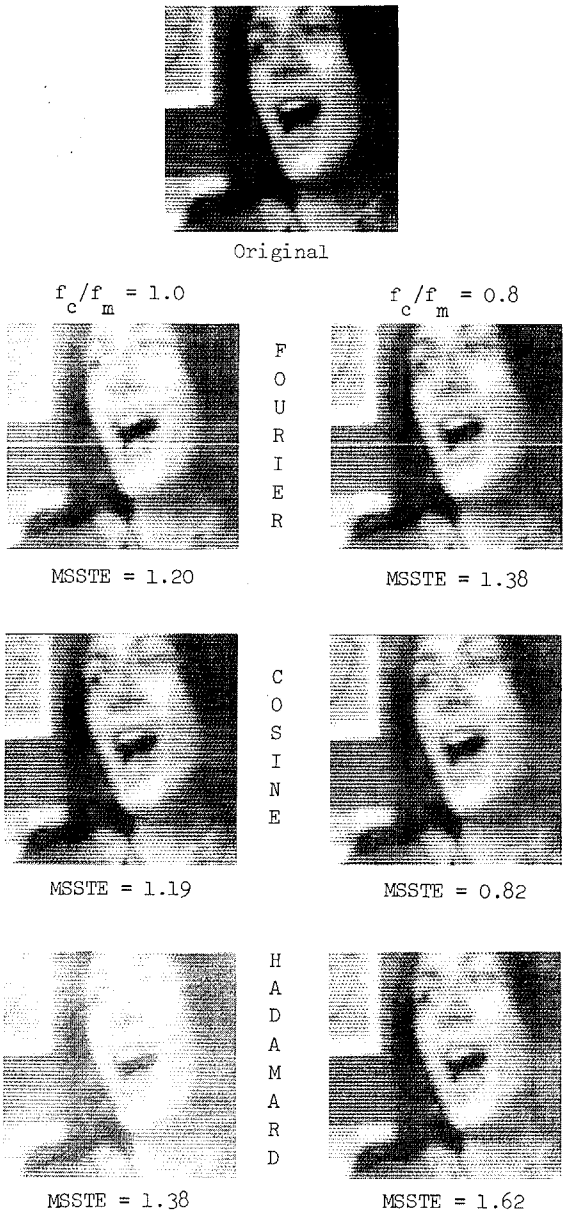

Fig. 5 - The original (8 bits/pel) and the processed images (1.1 bits/pel).