# REDUCTION OF BIT RATE FOR NOISY SEQUENCES BY THE INTEGRATION OF A MARKOV MOTION DETECTION ALGORITHM IN MPEG4 CODEC

*Khalil Hachicha, David Faura, Patrick Garda*

University Pierre and Marie Curie
4 place Jussieu 75252 Paris Cedex 05
Laboratories of Instruments and Systems at Ile de France
khalil.hachicha@lis.jussieu.fr

**ABSTRACT**

The objective of this study is to create a smart filter for the residue resulting from the difference between a matching macro block in two frames: the frame coded in the predictive mode and the reference frame. The filter is modeled by an algorithm of motion detection based on Markov Random Fields which reduces the variation of the luminance that more often results from noise than from an actual motion occurring between matching macro blocks. Thus, the bit rate decreases without changing the value of quantification and the quality of the sequence is maintained.

## 1. INTRODUCTION

A good video coder should efficiently compress the video, producing a low bit stream without sacrificing the video quality. Unfortunately when the sequence noise increases, the quantity of information necessary to transmit increases, thus increasing the bit rate. A solution to this problem is to change the value of quantification, which causes the image quality to dramatically decrease.

To resolve this problem we focus more specifically on a recent technique introduced by Alice Caplier [1] which relies on robust moving pixel segmentation and is capable of withstanding light variations while diminishing drag effects and noise. Unluckily the application of this algorithm is limited to quasi-fixed camera. If the camera does not remain in a mostly fixed position, a variation of luminance in the frame of difference, not related to the motion, is generated and thus the algorithm does not properly filter. However, with the accurate estimation/compensation process in the MPEG4 algorithm, the residue between the best match macro block in the reference frame and the current macro block represents roughly the variation of luminance directly related to the motion that occurs between the matching macro blocks and not the motion of the camera. Using this principle, we propose to include the application of the Markov theory for all types of sequences to detect robust motion between corresponding macro blocks in order to reduce the variation of luminance that may result from noise and thus the bit rate without changing the value of quantification.

In this work we present an MPEG4 basic algorithm, explain the fundamentals of the algorithm of motion detection based on the Markov, and describe the different steps that we followed to embed it in to our MPEG4 algorithm. We have thus tried to evaluate the new technique and assess its true performance with regard to different types of videos. We also evaluate the different gains in bit rate and the resulting image quality.

## 2. DESCRIPTION OF OUR MPEG4 CODEC

MPEG4 is the international standard established by the MPEG group in relation to moving picture encoding. The first step of this work, accelerated with the help of the MoMusys code proposed by ISO[2], was to develop an MPEG4 basic algorithm. Our codec supports 2 modes for encoding pictures: Intra coded (***I***) and predicted coded (***P***). The algorithm functions by subdividing the pictures into units called macro blocks (4 blocks of 8x8 luminance pixels).

***I*** pictures are coded without reference by performing discrete cosine transform (DCT), quantification and DC /AC prediction. They are then further compressed by variable length encoding. For ***P*** pictures, the movement between that picture and a previously encoded and decoded reference picture is investigated. The macro blocks are then taken one at a time to determine if there exists a match in the previous frame (the search area is +/-16 pixels). A SAD value is then obtained reflecting the degree of the fit for that particular block to a certain position in the reference frame. The algorithm then draws one of two conclusions. If the residue resulting from two macro blocks surpasses a certain threshold, the algorithm

concludes that the macro block is new to the video and it is thus coded in intra mode. If the difference between the macro blocks remains below the threshold, the algorithm accepts the best, though slightly imperfect, match.
In this case, the next step consists of improving the process of motion estimation in raising our research to the level of half-pixels.

We first attempt to find the best corresponding current macro block in the reference which will be interpolated. Then we search for the best corresponding block for each of the four blocks within the current macro block in the reference frame. In comparing the residue discovered by the two methods described above, we decided to transmit 1Mv for the complete macro block or 1Mv for each block in the macro block. This helps the decoder in reconstructing the current frame by finding the appropriate portions within the previously decoded frame. The combination of the motion vectors and the previous frame allows creating a motion compensated picture. The difference between the current and the compensated frame is then converted by DCT, quantized, variable length encoded together with related motion vectors information, and outputted as a bit stream. After undergoing inverse quantization and discrete cosine inversion, the quantized differential picture becomes the next reference picture stored in the memory [3][4].

## 3. MOTION DETECTION ALGORITHM BASED ON MARKOV MODEL

The modeling of the Markov, from a statistical point of view, relies upon the estimation of the ensemble of contextual information which permits local decision making. The decision making process thus depends not only on the available data for the point, but also on the information received from neighboring models which are either spatial or temporal. We have reworked a model using the potential functions foreseen by the detection of motion combining the spatial and temporal information[6]. It is composed of two distinct steps: the first consists of a pretreatment phase through which the variance is determined. The absolute value of the difference matrix is then calculated and binarized by setting a threshold.

The second grouping algorithm of the ICM for updating the binary state of the pixels of difference (moving or not) which is made site by site in the sense that every change in state is taken immediately into account in the relaxation of the neighboring site. In this way it will allow the convergence to the first minimum of the energy function. In order to calculate the energy, one must know the state of the pixels belonging to a neighborhood defined by eight spatial neighbors and two temporal neighbors.

The energy expression consists of:

- The energy associated with the data :

$$Ud(s) = \frac{1}{2\sigma^2}(0(s) - \psi(s))^2 \begin{cases} \psi(s) = 0 \text{ si s} = 0 \\ \psi(s) = \alpha \text{ si s} = 1 \end{cases} \quad (1)$$

$$\begin{cases} \text{O(s)} = \psi\text{(s)} + \text{b} \\ \psi\text{: Modeling the observations} \\ \text{b: White noise with a variance } \sigma^2 \end{cases}$$

- The energy associated with the model (2):

This energy consists of the spatial energy (3) that is supposed to model the consistency and the compactness of a moving object and the temporal energy (4) which represents the variation of the intensity function when the frame changes.

$$Um(s, rf, rp) = Us(s) + Ut(s, rp, rf) \quad (2)$$

$$\text{Us(s)} = \sum_s \text{Vs(s)} \quad \text{With} \begin{cases} Vs = -\beta\text{s si s} = \text{r} \\ \text{Vs} = +\beta\text{s si s} \neq \text{r} \end{cases} \quad (3)$$

$$\text{Ut(s, rf, rp)} = \text{Vp(s, rp)} + \text{Vf(s, rf)} \quad (4)$$

$$\begin{cases} \text{Vp} = -\beta\text{p si s} = \text{rp} \\ \text{Vp} = +\beta\text{p si s} \neq \text{rp} \end{cases} \quad \begin{cases} \text{Vf} = -\beta\text{f si s} = \text{rf} \\ \text{Vf} = +\beta\text{f si s} \neq \text{rf} \end{cases}$$

For each image site, the local energy is calculated relative to both the immobile state and the mobile state and we allocate the state which minimizes the energy to the site being treated. The minimization of energy has a filtering effect on the noise and the partial reconstruction of the moving zones. Leaving ICM, we achieve an image of minimal energy which represents the binary map motion. The product between this binary map motion and the differences between the images generate an image of varying luminance filtered by the reconstruction of some zones allocated by the motion.

The parameters of the algorithm of motion detection are chosen as follow[6][7][8]:

$$\beta p = 10, \beta s = 20, \beta f = 30, \alpha = 10 \quad (5)$$

Following the study and the development of this algorithm, we performed tests on several sequences in order to evaluate the effectiveness of the technique on different images. Some of our results can be visualized below (fig.1). In the first we find simple motion detection by a difference between two consecutives frames.

The second represents the binary motion map created from the frame of differences. The multiplication by the mask allows only the conservation of the variation of luminance which reflects the motion.

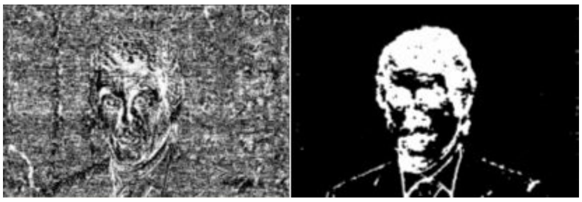

Fig.1 (a) Frame of difference (b) Markov Mask Motion

## 4. ALGORITHMIC INTEGRATION IN OUR MPEG4 CODEC

The coding scheme can be represented as follows (fig.2):

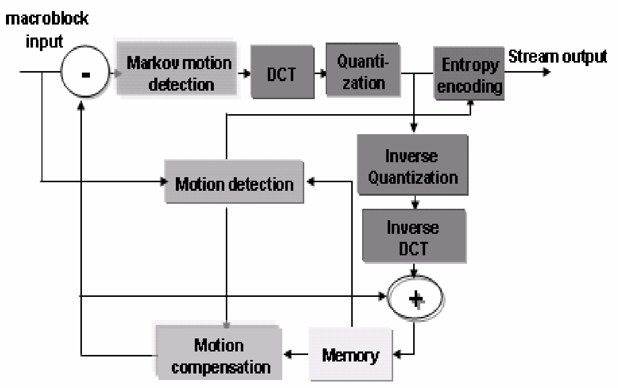


Fig.2

The algorithm encode a prediction error only for the pixels for which the motion mask information is set to 1. The mask is not to code. To create a Markov motion mask for each macro block, one must first know its type of coding which is decided by the process of motion estimation. There exist two possibilities: first if the macro block is Intra coded then we will set every filtration value within the masqued binary image to “1”. In other words, all of the information contained in the macro block of difference will be coded without any filtering operation. The second possibility is that the macro block is Inter coded and thus there exists two additional possibilities. The first is to send one motion vector per macro block (1 Mv/Mb) to the coder, in which case a mask for the entire macro block will be created. The second is to send four motion vectors per macro block (4 Mv/Mb) and create a mask for each block. The creation of a mask is based on the energy calculation as we discussed previously. It is important to note the following point when creating the binary masqued image. First, we will restrict our energy calculations to the spatial and data energies. The temporal energy is unknown because the state of the neighboring pixels at t-1 and t+1 is not known. Second, the value of difference of the matching pixels located on the macro block border will remain the same as it is impossible to know the value of difference of the spatial neighbors. When the binary mask is created, it is multiplied by the macro block of difference, thus “cleaning” it. The synopsis of the following methodology is described in detail in the flowchart below (fig.3):

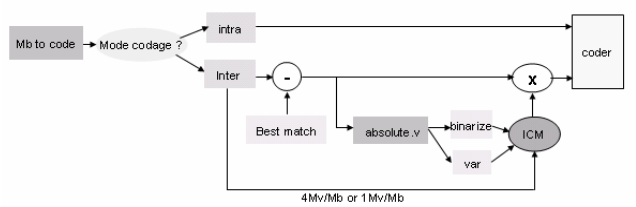


Fig.3

## 5. RESULTS AND INTERPRETATIONS

Our tests were limited to the small formats QCIF video. We choose two sequences: foreman and france3. We present (Table1) the different bit rates and PSNR obtained achieved in compressing the above mentioned videos with our codec MPEG 4 and both with and without Markov. Note that the value of quantization is held constant throughout each test (and is changed between tests).

| | **Test1** | **Test2** | **Test3** | **Test4** |
|---|---|---|---|---|
| **Foreman_qcif** | | | | |
| Bit rate | 317 | 186 | 128 | 98 |
| Psnr luminance | 37.83 | 35.3 | 33.7 | 32.53 |
| **Foreman_markov** | | | | |
| Bit rate | 282 | 170 | 121 | 94 |
| Psnr luminance | 36.26 | 34.6 | 33.34 | 32.33 |
| **France 3** | | | | |
| Bit rate | 291 | 181 | 130 | 102 |
| Psnr luminance | 38.83 | 36.26 | 34.51 | 33.22 |
| **France3markov** | | | | |
| Bit rate | 244 | 159 | 118 | 95 |
| Psnr luminance | 36.13 | 34.83 | 33.69 | 32.73 |

Table 1

The filtering of the macro blocks by the created masks permitted us to reduce the bit rate while conserving the same value of quantification. We have drawn the following conclusions:

For sequences that practically lack of great noise (foreman), the Markov Motion detection algorithm does not apply well. This is expected as the algorithm is not designed for this type of sequence.

For the second type of sequence: france3 (an excerpt of a news emission displaying a lot of motion and luminance variation), our algorithm exhibits a much better performance. The reduction of the PSNR doesn't reflect the image quality that we obtained which is closely comparable to this algorithm without applying Markov. However, at the same time, the bit rate is reduced in an interesting fashion. This results from the clever Markov filtering of the value of luminance variations between the matching macro block. Thus, our method becomes very interesting.

To study the influence of this technique on sequences displaying even more noise, we selected another france3 excerpt. The excerpt is noisier than the first and display rapid motion and strong luminance variation. Our results are listed below:

| | **Test 1** | **Test 2** | **Test 3** | **Test 4** |
|---|---|---|---|---|
| **Fr 3_reports** | | | | |
| Bit rate | 570 | 365 | 265 | 210 |
| Psnr luminance | 38.1 | 35.64 | 33.97 | 32.73 |
| **Fr3_reports_markov** | | | | |
| Bit rate | 480 | 310 | 235 | 190 |
| Psnr luminance | 35.71 | 33.57 | 32.62 | 31.85 |

Table 2

Using our achieved data, we can draw the following conclusion: the greater the noise in the sequence, the better the performance of our algorithm.

## 6. CONCLUSIONS AND OUTLOOK

Our work is organized in three parts: 1) the understanding and the development of a basic MPEG4 algorithm, 2) the study and application of the algorithm of motion detection based on Markov Random Fields, and 3) the integration of the Markov algorithm into our MPEG4 algorithm.

This work presents an effective solution in reducing the unnecessary information contained in macro blocks of difference. However, one must be note that the results are heavily dependent on the type of sequence.

The outlook for our work is to evaluate the algorithmic complexity introduced, integrate the motion detection algorithm into the new MPEG4 part 10 standards, and estimate its contribution to this standard.

## 7. REFERENCES

[1] A.Caplier, Modéle Markovien de detection de movement dans les sequences d'images: thèse institut national Polytechnique de Grenoble (INPG), france Décembre 1995

[2] ISO/IEC 14496-5:2001.Coding of Audio-Visual Objects – Part 5: Reference Software, 2d Edition, 2001

[3] ISO/IEC 14496-2:2001.Coding of Audio-Visual Objects Part2: Visual, 2d Edition, 2001.

[4] The MPEG4 Book, Fernando Pereira, Tourajdi Ebrahimi, p294-315, 2002 Pearson Education.

[5] S.-K. Weng, C.-M. Kuo and C.-H. Hsieh, "Motion estimation algorithm for image sequence coding," IEE Optical Engineering, vol. 36, no. 12, pp. 3272-80, Dec.1997.

[6] Frantz Lohier, Méthodologie de programmation et évaluation des processeurs de traitement de signal parallèles pour le traitement d'images en temps réel. Thèse LISIF, France février 2000

[7] C.Dumontier, Etude et mise en oeuvre temps réel d'un algorithme de detection de movement par approche markovienne. Thèse de Doctorat de l'Institue National Polytechnique de Grenoble (INPG), France, Novembre 1996

[8] P.Lalande, P.Bouthemy, Détection de movement dans les sequences d'image selon une approche markovienne. Doctorat de l'université de Rennes I, France 1990

[9] F.Lohier, P.Garda, L.Lacassagne Procedé et dispositif de traitement de sequences d'images avec masquage.
Brevet UPMC, Francais en cours d'extension Internationale demande FR 6206L du 3 fevrier 2000.