Iterative joint source-channel decoding of variable length encoded video sequences exploiting source semantics

Hang Nguyen, Pierre Duhamel
Alcatel, Research and Invovation Department, Route de Nozay, F91460, FRANCE (hang.nguyen@alcatel.fr)
CNRS/LSS, Supelec, Plateau de Moulon, F92190, FRANCE (pierre.duhamel@lss. supelec.fr)

Abstract—**This paper proposes an iterative joint source-channel decoding of variable length codes (VLC) sequences exploiting the inherent redundancy in the source stream, namely (i) the VLC structure (ii) the sequence structure. An optimum Soft-Input Soft-Output (SISO) Maximum Likelihood (ML) decoder of VLC sequences which is able to exploit both types of redundancies is first proposed, and shown to be optimal. This algorithm is then used in an iterative joint source-channel decoding. Performance results for transmitting video over AWGN channels are presented and compared to the Soft-Output Viterbi Algorithm (SOVA) VLC decoder.**

I. INTRODUCTION

Joint source-channel decoding (JSCD) aims at making use of the bit stream properties to introduce robustness in the VLC decoding. Three main categories of JSCD can be distinguished depending on the the bit stream properties. The first family [2-5] uses the remaining source redundancy due to the VLC structure. The second family [6-8] uses redundancy associated to both the VLC structure and the Markov model of the source which exploits the correlation between two successive source samples. The third family exploits the structure of both the VLC and the bit stream. However, first results of this third category [9] only exploit one part of the structural properties of the bit stream. We have shown in a previous paper [10] that if additional structural properties of the bit stream, such as these constraints, are exploited, much more redundancy can be used for correcting errors. In H.263 standard[1], each VLC codeword corresponds to a triplet (*run, level, last*). Papers [10,11,15] showed that any compressed image block sequence S should meet the constraints:

1. Sum of "run" values over all VLC codewords (denoted the ***number of DCT coefficients***) should be less or equal to the total number of coefficients $N_{DCTcoef}$

$$R_S = \sum_{VLC\ codewords\ of\ sequence\ S} (run_{vlc} + 1) \leq N_{DCTcoef} \quad (1)$$

2. Only the last codeword has the field "last" equal to one.

Similar properties exist in H.263++, H.264, MPEG standards. Sequences meeting these constraints will be denoted as ***feasible sequences***. However, a straightforward modification of the SISO algorithms of [2-9] could hardly take into account these constraints on the whole sequence issued from the source. An optimum hard-output algorithm ables to exploit these additional properties is proposed in [11] and a reduced complexity version in [15]. The classical way of obtaining Soft-Output Viterbi Algorithm (SOVA) for VLC decoding [3,14] could be applied to the algorithm of ref.[11]. Nevertheless, this version would involve only the survivors at each step. Our proposed version involves an average **over all feasible paths**, thus reaching **optimality**. The corresponding decoder is described in the first part of the paper. It delivers both optimum hard-outputs and optimum soft-outputs in the ML sense. The proposed optimal SISO VLC source decoder is then applied in an iterative joint source-channel decoding scheme on real video sequences.

II. OPTIMAL ML SOFT-OUTPUT DEFINITION

A. *Problem formulation and definition*

With same notations as [10,11,15], Y denotes the received image block sequence of N bits; Ω : the set of all feasible sequences; X an element of Ω . If the *a priori* distribution of the source is uniform, the optimum hard-output sequence in the maximum likelihood sense is defined by*:*

$$X_{ML} = \arg \underset{X \in \Omega}{Max}\, P(Y/X) = \arg \underset{X \in \Omega}{Max}\, P(X/Y) \quad (2)$$

To get the optimum soft-output solution, all feasible sequences of the set Ω have to be considered. We note the binary bit streams of X, X^{ML} and Y as:

$$X = x_1 ... x_i ... x_N\ ,\ Y = y_1 ... y_i ... y_N\ ,\ X^{ML} = x_1^{ML} ... x_i^{ML} ... x_N^{ML}\ .$$

The optimal reliability (soft-output) associated to the *i-th* bit x_i^{ML} of the binary bit stream of X^{ML} as: $L(x_i^{ML})$ is:

$$L(x_i^{ML}) = \ln\left(\frac{P(x_i = x_i^{ML})}{P(x_i \neq x_i^{ML})}\right) = \ln\left(\frac{\sum_{X \in \Omega \,|\, x_i = x_i^{ML}} P(X|Y)}{\sum_{X \in \Omega \,|\, x_i \neq x_i^{ML}} P(X|Y)}\right) \quad (3)$$

We partition the set Ω with respect to the value (± 1) of x_i , into the sets: $\Omega_i^{(\pm 1)} = \{X \in \Omega | x_i = \pm 1\}$. The ***marginal probabilities*** of x_i is:

$$P(x_i = \pm 1 | Y) = \sum_{X \in \Omega_i^{(\pm 1)}} \prod_{l=1}^{N} p(y_l | x_l) P_a(x_l) \quad (4).$$

If $P(x_i = \pm 1 | Y)$ and X^{ML} are known, the soft-outputs defined by equation (3) can be computed.

B. *Proposed optimal ML SISO VLC decoder outlines*

The proposed SISO VLC decoding algorithm is based on the hard-output VLC decoding algorithm of [11,15]. The survivor selection takes into account the conventional Viterbi metric[13], the VLC structure and the source semantics constraints on the sequence [10,15]. Only candidate sequences of same length λ in bits and same number of DCT coefficients r are considered for the survivor selection. The set of all feasible paths associated to each possible value of (r, λ) is denoted $\Delta''_{\lambda,r}$. Only one ML survivor $S^{ML}_{\Delta''_{\lambda,r}}$ is selected and all other sequences of the set

0-7803-8554-3/04/$20.00

$\Delta"_{\lambda,r}$ are discarded. Hence, the soft-outputs should take into account this decision of saving only one survivor path and discarding all other paths. **The information on the reliability of this decision** is stored into two vectors, denoted as ***decision reliability*** vectors $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$, which are computed from the probabilities of all associated paths including the discarded ones. The *decision reliability* vectors correspond to the marginal a posteriori probabilities of the set $\Delta"_{\lambda,r}$. The two *decision reliability* vectors $Mg^{\pm 1}(S^{ML}_f)$ associated to the final survivor S^{ML}_f (the hard-output solution) of length N provides the values of the marginal probabilities $P(x_i = \pm 1 | Y)$, and the soft-outputs defined by equation (3). The soft-outputs provided by existing SOVA VLC decoders [2-5,14] are computed from only one part of the set Ω, hence they are not optimal because some of the discarded paths are not taken into account. The soft-outputs provided by our algorithm include the whole set Ω, hence are optimal. In addition, they exploit both types of source redundancy associated to the structures of the VLC and of the sequence. This makes them more efficient than the existing SOVA-based soft-outputs (section V).

III. SOFT-OUTPUT COMPUTATION

A. *Algorithm description outlines*

Sequences of length strictly smaller than N bits are called ***incomplete sequences***. The algorithm is based on the recursive computation of a number of lists of sequences L_k which contains all ***incomplete feasible*** VLC survivor paths of k codewords. At each recursion, L_{k-1} is available, L_k is to be constructed. For each possible length λ of sequences of the list L_k, we compute the set Δ'_λ of ***feasible*** sequences of length λ,and being the concatenation of one sequence from L_{k-1} and one codeword from the VLC codebook. We identify the ML sequence $S^{ML}_{\Delta'_\lambda}$ over the set Δ'_λ, and its associated number of DCT coefficients $r_{S^{ML}_{\Delta'\lambda}}$. Then, we compute the subset $\Delta"_{\lambda,r}$ of the set Δ'_λ of sequences whose associated numbers of DCT coefficients are equal to r which is smaller than $r_{S^{ML}_{\Delta'\lambda}}$. We identify and save the ML sequence $S^{ML}_{\Delta"_{\lambda,r}}$ over each set $\Delta"_{\lambda,r}$. The two associated *decision reliability* vectors $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ are computed. At this step, a list of survivor feasible sequences for all possible associated values r smaller than $r_{S^{ML}_{\Delta'\lambda}}$, containing exactly k VLC codewords, and of length λ, is obtained. If $\lambda = N$, these survivors are stored in F. If $\lambda < N$, these survivors are stored in L_k. The recursion is finished when all survivor sequences are of length N. At the end, we obtain the list F containing all feasible VLC survivor sequences of k codewords, of length equal to N bits, and meeting both VLC structure constraints and source-induced constraints on the whole sequence.

B. *Computation of the decision reliability vectors*

This section outlines the computation of the elements $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}}, i)$ of the two *decision reliability* vectors $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ associated to the ML sequence $S^{ML}_{\Delta"_{\lambda,r}}$ mentioned in the previous section.

At step 2.6, the ML sequence $S^{ML}_{\Delta"_{\lambda,r}}$ associated to the Viterbi metrics over the set $\Delta"_{\lambda,r}$ is identified. Only this survivor sequence $S^{ML}_{\Delta"_{\lambda,r}}$ is saved, while the other sequences of the set $\Delta"_{\lambda,r}$ are discarded. Thus, the two vectors $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ should store **the information on the reliability of this decision** of selecting one sequence as the survivor and of discarding the other sequences. More precisely, the quantities $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ are **the (non normalized) marginal a posteriori probabilities of each symbol.** In this paper, we consider binary sequences.

Hence, for each bit to be decoded, we need to compute the probabilities of this bit being equal to 1 or -1. The set Ω of all feasible sequences (section III-A), is required for computing and normalizing these probabilities. But, the whole set Ω is known only at the end of the algorithm. Thus, in the algorithm, we need to recursively compute and store the *marginal and non-normalized a posteriori probabilities* $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ *of each bit of being equal to 1 and –1*, of the set $\Delta"_{\lambda,r}$. A normalization operation is made at the end of the algorithm (section III-F).

The impact of all sequences $S_{\Delta"_{\lambda,r}}$ of the set $\Delta"_{\lambda,r}$ (discarded or not in the survivor selection process) is taken into account by summing the contributions $\psi^{\pm 1}(S_{\Delta"_{\lambda,r}})$ of each candidate sequence:

$$Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}}) = \sum_{S_{\Delta"_{\lambda,r}} \in \Delta"_{\lambda,r}} \psi^{\pm 1}(S_{\Delta"_{\lambda,r}}).$$

For binary sequences, the quantities $\psi^{\pm 1}(S_{\Delta"_{\lambda,r}})$ are *the marginal and non-normalized a posteriori probabilities* of each bit of being equal to 1 and –1, of the set of all discarded paths corresponding to the selection of the sequence $S_{\Delta"_{\lambda,r}}$ as survivor sequence.

Hence, the quantities $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ and $\psi^{\pm 1}(S_{\Delta"_{\lambda,r}})$ defined above, are vectors of size λ, corresponding to λ bits of the sequences of $\Delta"_{\lambda,r}$. Hence, the λ components of the vectors $Mg^{\pm 1}(S^{ML}_{\Delta"_{\lambda,r}})$ are computed by: $Mg^{\pm1}(S^{ML}_{\Delta"_{\lambda,r}},i)=\sum_{S_{\Delta"_{\lambda,r}}\in\Delta"_{\lambda,r}} \psi^{\pm1}(S_{\Delta"_{\lambda,r}},i)$.

We now evaluate the components of the vectors $\psi^{\pm1}(S_{\Delta"_{\lambda,r}})$. At the construction of the list L_k, all sequences $S_{\Delta"_{\lambda,r}}$ of the set $\Delta"_{\lambda,r}$ are the concatenation of one sequence $S^{L_{k-1}}_{\Delta"_{\lambda,r}}$ from the previous list L_{k-1} and one VLC codeword $V^k_{\Delta"_{\lambda,r}}$ from the codebook: $S_{\Delta"_{\lambda,r}}=S^{L_{k-1}}_{\Delta"_{\lambda,r}}V^k_{\Delta"_{\lambda,r}}$.

We note:

- s_i^{ML} (resp. $s_i^{\Delta"_{\lambda,r}}$) the *i*-th bit of sequence $S^{ML}_{\Delta"_{\lambda,r}}$ (resp. $S_{\Delta"_{\lambda,r}}$),
- μ and ν, respectively, the lengths in bits of the subsequence $S^{L_{k-1}}_{\Delta"_{\lambda,r}}$ and of the codeword $V^k_{\Delta"_{\lambda,r}}$,
- $Mg^{\pm1}(S^{L_{k-1}}_{\Delta"_{\lambda,r}})$ are the two associated *decision reliability* vectors of $S^{L_{k-1}}_{\Delta"_{\lambda,r}}$.

The μ first components of $\psi^{\pm1}(S_{\Delta"_{\lambda,r}})$ corresponding to the μ bits of the sub-sequence $S^{L_{k-1}}_{\Delta"_{\lambda,r}}$ are equal to the associated values taken from $Mg^{\pm1}(S^{L_{k-1}}_{\Delta"_{\lambda,r}})$ and multiplied by the probability of $V^k_{\Delta"_{\lambda,r}}$ in order to take into account the concatenation:

$$\forall i\in[1,\mu]:\ \psi^{\pm1}(S_{\Delta"_{\lambda,r}},i)=Mg^{\pm1}(S^{L_{k-1}}_{\Delta"_{\lambda,r}},i)*P(V^k_{\Delta"_{\lambda,r}}).$$

The ν last components of $\psi^{\pm1}(S_{\Delta"_{\lambda,r}})$ corresponding to the ν bits of the codeword $V^k_{\Delta"_{\lambda,r}}$ are computed as follows. If the corresponding bit $s_i^{\Delta"_{\lambda,r}}$ of the codeword $V^k_{\Delta"_{\lambda,r}}$ is equal to 1, there is no contribution of the sequence $S_{\Delta"_{\lambda,r}}$ for $Mg^{-1}(S^{ML}_{\Delta"_{\lambda,r}})$, thus: $\psi^{-1}(S_{\Delta"_{\lambda,r}},i)=0$; and the contribution of the sequence $S_{\Delta"_{\lambda,r}}$ for $Mg^{+1}(S^{ML}_{\Delta"_{\lambda,r}})$ is:

$$\psi^{+1}(S_{\Delta"_{\lambda,r}},i)=(Mg^1(S^{L_{k-1}}_{\Delta"_{\lambda,r}},\mu)+Mg^{-1}(S^{L_{k-1}}_{\Delta"_{\lambda,r}},\mu))*P(V^k_{\Delta"_{\lambda,r}})$$

Similarly, if $s_i^{\Delta"_{\lambda,r}}=-1$, $\psi^{+1}(S_{\Delta"_{\lambda,r}},i)=0$, then:

$$\psi^{-1}(S_{\Delta"_{\lambda,r}},i)=(Mg^1(S^{L_{k-1}}_{\Delta"_{\lambda,r}},\mu)+Mg^{-1}(S^{L_{k-1}}_{\Delta"_{\lambda,r}},\mu))*P(V^k_{\Delta"_{\lambda,r}})$$

C. *Final soft-output computation*

At the end of the recursion, the list F is obtained. Each sequence S_F in the list F has two associated vectors of non-normalized a posteriori probabilities $Mg^{\pm1}(S_F)$. The optimal hard-output sequence S_f^{ML} is selected from the obtained list F as the survivor sequence with the smallest metric[11,13]. The associated vectors $Mg^{\pm1}(S_f^{ML})$ are computed as: $Mg^{\pm1}(S_f^{ML},i)=\sum_{S_F\in F} Mg^{\pm1}(S_F,i)$ and the probabilities $P(x_i=\pm1|Y)$ are obtained by normalizing the vectors $Mg^{\pm1}(S_f^{ML})$ as:

$$P(x_i=\pm1|Y)=\frac{Mg^{\pm1}(S_f^{ML},i)}{Mg^{1}(S_f^{ML},i)+Mg^{-1}(S_f^{ML},i)} \quad (5).$$

The corresponding log-likelihood can be computed as:

$$L(x_i^{ML})=P(x_i=\pm1|Y) \text{ if } x_i^{ML}=\pm1.$$

D. *On the optimality of the algorithm*

The algorithm provides the same hard-output as ref.[11] that has been proved to be optimal. The optimality of the soft-outputs as defined in section III can be proved by a recursive proof that is outlined in this section. As defined in section III-A, the soft-ouputs are optimum if the whole set Ω is taken into account. We denote Ω_k the subset of the set Ω of sequences of *k* codewords, and *K* the set of values *k* such as Ω_k is not empty. $\{\Omega_k\}_{k\in K}$ is a partition of the set Ω. It can be proven that for each value *k* of the set *K,* there is one and only one sequence of *k* codewords in the list *F*. It can be also proven that the *decision reliability* vectors $Mg^{\pm1}$ of each sequence of the final list *F* correspond to the marginal a posteriori probabilities of the associated set Ω_k. Hence, the vectors $Mg^{\pm1}(S_f^{ML})$ associated to the final survivor are the marginal a posteriori probabilities of the whole set Ω. Thus, all the paths are taken into account and the optimality is reached.

E. *On the complexity of the algorithm*

Like the SOVA for VLC[3], this algorithm performs an update of the soft-outputs associated to each selected survivor path. However, the SOVA is usually based on a log-max approximation. Hence, each update operation corresponds to some comparisons and some additions. Each update operation of our algorithm corresponds to the computation of the *decision reliability* vectors as described in section IV-A. More precisely, for each of the two *decision reliability* vectors, the update requires one addition and one multiplication, or only one multiplication, or no operation, depending on the considered case. Nevertheless, our algorithm is optimal. A log-max approximation can be applied to reduce the complexity, and specially the multiplications. In this case, this log-max algorithm has a comparable complexity with respect to the SOVA for VLC[3]. Our algorithm has another advantage: more paths are pruned than in the existing algorithms, thanks to the exploitation of the source semantics. Fewer paths imply fewer required computations.

V. SIMULATION RESULTS

As in [10,11,15], the performance metric is chosen as the image block error rate (IBER), defined as follows :

$$r = \frac{number\ of\ blocks\ that\ are\ erroneously\ decoded}{number\ of\ transmitted\ blocks}$$

Our intent here is only to show how these new soft-outputs and the full use of the source semantics could improve the decoding performance. A set of H.263 encoded image blocks from three classical video sequences: “Mother-daughter”, “Foreman”, “Irene”, is used for simulation. This set of image blocks is transmitted over a Gaussian (AGWN) channel with the BPSK modulation, then decoded with two iterative joint source-channel decoding schemes: (i) using the conventional SOVA VLC source decoder exploiting only the VLC-structure[2-5], (ii) using the proposed optimal SISO VLC source decoder exploiting both types of redundancy: from the VLC-structure and from the structure of the sequence. The iterative joint source-channel decoding algorithm is shown in Fig.1. The quantities exchanged between the channel and source decoders are extrinsic probabilities. Fig. 2 plots the IBER at the output of the channel decoder.

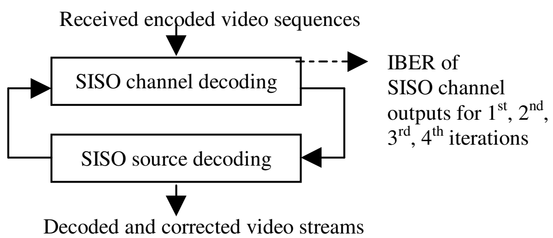

Fig.1: Iterative joint source-channel decoding

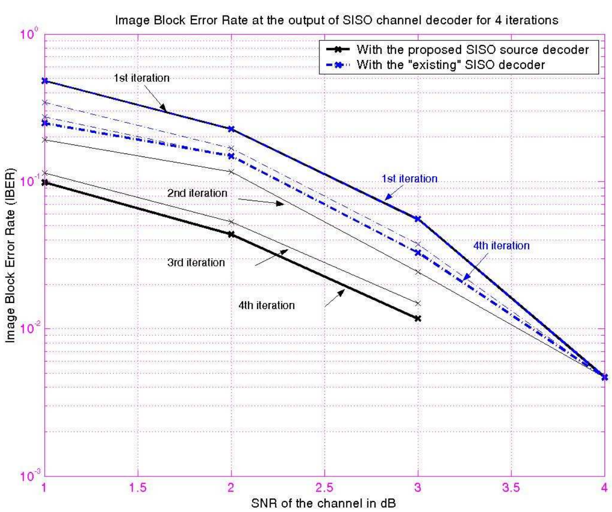

Fig.2: IBER at the output of the channel decoder for two iterative joint source-channel decoding schemes: (i) using the SOVA VLC decoder exploiting only the VLC structure (ii) using the proposed SISO VLC decoder exploiting both the VLC structure and the sequence structure

In this iterative decoding framework, our algorithm outperforms the SOVA decoder in a noticeable manner on the whole range of SNR and for any iteration. A gain of 0.5 to 2.3 dB in terms of IBER is obtained. Fig.2 shows that at the 1st iteration the performance of the channel decoding is the same in both cases, because the data have not been processed by the SISO source decoder yet. However, after the first iteration, the iterative system using our SISO decoder outperforms already the one using the SOVA VLC decoder at its 4th iteration, which is also its limit iterative performance. This gain grows even more significantly at the 2nd iteration with our proposed SISO VLC decoder. The gain obtained by the iterations (between the 1st and the 4th iterations) is about 0.5 to 1.5 dB in terms of IBER for our VLC decoder, while only up to 0.5 dB for the SOVA VLC decoder. The gain introduced by iterations is much larger in our case because our source decoder is more efficient. For the same performance, our SISO source decoder only needs one iteration to reach the limit performances of the previous algorithms. Hence**,** the complexity is roughly divided by 4 for the same performance.

VI. CONCLUSION

This paper proposes a new SISO VLC decoder which can deliver both optimum hard-outputs and optimum soft-outputs in the ML sense, with linear complexity in the number of VLC codewords. This decoder can exploit both the VLC structure and the intrinsic structure of feasible VLC sequences of image blocks. The soft-outputs provided by the proposed algorithm outperform the ones from the conventional SOVA, while the complexity is similar to that of existing methods for decoding VLC sequences, which only make use of a partial redundancy. Furthermore, our method always provides a feasible VLC sequence of a valid image block.

REFERENCES

[1] *ITU-T Recommendation H.263 :* 03-96, *H263+ :* 02-98, *H264*.
[2] R.Bauer, J. Hagenauer, “ On Variable Length Codes for Iterative Source-Channel Decoding”, *Proc. of IEEE DCC*, 2001.
[3] Bauer, Hagenauer,“Iterative Source-Channel Decoding based on a Trellis representation for Variable Length Codes”,*ISIT* 2000.
[4] J.Wen, J.D.Villasenor, “ Utilizing soft information in decoding variable length codes”, *Proc. IEEE DCC*, March 1999.
[5] L.Guivarch, J.C. Carlach, P. Siohan, “Joint source-channel soft decoding of Huffman codes with Turbo-codes*”*, *DCC* 2000.
[6] Subbalakshmi, Vaisey, “On the Joint Source-Channel Decoding of Variable-Length Encoded Sources: the Additive Markov Channel Case”, *IEEE Trans. On Com.,* Sept.2003.
[7] Park, Miller, “Joint Source-Channel Channel Decoding of Variable-Length Encoded Data by Exact and Approximate MAP Sequence Estimation”, *IEEE Trans. On Com.*, Jan.2000.
[8] Sayood, Otu, Demir, “Joint Source-Channel Channel Coding for Variable-Length Codes”, *IEEE Trans. On Com.*, May 2000.
[9] S.Kaiser, M.Bystrom, “Soft decoding of variable length codes”, *IEEE ICC*, 2000, volume 3.
[10] H. Nguyen, P. Duhamel “Estimation of Redundancy in Compressed Image and Video Data for Joint Source-Channel Decoding”, *IEEE Globecom,* Dec 2003.
[11] H. Nguyen, P. Duhamel, Jerome Brouet “Optimal VLC Sequence Decoding Based on Compressed Image and Video Stream Properties”, *ICASSP,* May 2004.
[12] Hashimoto, “A list-type reduced-constraint generalization of the Viterbi algorithm*”*, *IEEE Trans. On Inf. Theory*, 11/1987
[13] Forney, “The Viterbi Algorithm*”*, *Proc. of the IEEE*, 03/1973.
[14] Hoeher, “Advances in Soft-Output Decoding”, *IEEE GLOBECOM*, vol.2, Page(s): 793 –797, 29 Nov-2 Dec 1993.
[15] H. Nguyen, P. Duhamel ,J.Brouet,D.Rouffet “Robust VLC sequence decoding exploiting additional video stream properties with reduced complexity”, *IEEE International Conference on Multimedia and Exposition* (*ICME*), June-July 2004