Journal of Southeast University (English Edition) Sept. 2002 Vol.18 No.3 ISSN 1003—7985

Soft Decoding Scheme of Convolution Code Combined with Huffman Coding*

Guo Dongliang** Chen Xiaoqiang Wu Lenan

(Department of Radio Engineering, Southeast University, Nanjing 210096, China)

Abstract: This paper proposes a modification of the soft output Viterbi decoding algorithm (SOVA) which combines convolution code with Huffman coding. The idea is to extract the bit probability information from the Huffman coding and use it to compute the *a priori* source information which can be used when the channel environment is bad. The suggested scheme does not require changes on the transmitter side. Compared with separate decoding systems, the gain in signal to noise ratio is about 0.5 – 1.0 dB with a limited added complexity. Simulation results show that the suggested algorithm is effective.

Key words: soft output Viterbi decoding, *a priori* information, Huffman coding, convolution code

Convolution codes are widely applied in communication system due to its outstanding performance. The commonly used decoding algorithms include the Viterbi algorithm, sequence decoding and threshold algorithm. Among them the Viterbi algorithm possesses the best performance, especially in the short constraint length case. SOVA (soft output Viterbi algorithm) can achieve better performance than the hard decision Viterbi algorithm, which has been proved that 3-bit SOVA can improve the performance by 2 dB in terms of AWGN channel.

Hagenauer presented an improved decoding algorithm of convolution code in 1995[1]. In this algorithm it was supposed that there was some correlation between adjacent bits and the output of the source coder. This correlation was expressed by the likelihood value of *a priori/posteriori* probabilities[2] and an improved SOVA decoding algorithm was provided and better performance was achieved. This algorithm is suitable for the fixed frame bits, for example, the PCM coding bits.

In this paper we apply the idea of Ref.[3] to VLC (variable length codes) and take advantage of their high compression capacity. Huffman code is one of the most important VLC, which constructs the shortest mean length code-words only due to the symbol probabilities and is also declared as the optimum coding. Huffman code provides the symbol probabilities and SOVA provides the trellis structure and soft decoding value. From the symbol probabilities we can obtain the bit probabilities and then the *a priori* information of source. When the difference of meshes belonging to the two competing paths is large, the Viterbi decoding is reliable, otherwise we should decode according to not only the channel information, but also the source information. The threshold value of the suggested algorithm can decrease the computational complexity and make the system more intelligent. The simulation results show that the gain in SNR (signal to noise ratio) is 0.5 – 1.0 dB with a limited additional complexity and more improvement of performance can be gained when the channel SNR decreases.

1 Decoding Criterion

SOVA belongs to the maximum likelihood decoding algorithm whose idea is to find the most likely information path that generates the received sequence. Suppose the information sequence M is encoded into sequence X, and then is transmitted into a noisy channel where the received sequence is denoted as Y. If the sequence M' from the decoder is not equal to the information sequence M, the decoder has not corrected the sequence deteriorated by channel noise. Suppose all the possible information sequences have the same probability. When the sequence Y is received, if $\log P(Y|X(M')) \geqslant \log P(Y|X(M)), M' \neq M$, the sequence is decoded as M'. The probability $\log P(Y|X)$ is the logarithm likelihood function.

The above algorithm supposes that all the information sequences have equal probabilities, which is inconsistent with the source property. If we take the information sequence probabilities into account, when

Received 2002-03-07.

* The project supported by the National Natural Science Foundation of China (60072013).

** Born in 1976, male, graduate.

the sequence Y is received, we should find the sequence X which maximizes the probability $P(X|Y)$.

Using Bayes rule, $P(X|Y)P(Y) = P(Y|X) \cdot P(X)$, the final criterion, taking the source information into account, becomes

$$\max_X \left[\log P(Y|X) + \log P(X) \right] \tag{1}$$

The first term of (1) corresponds to channel information and the second one is *a priori* information about the source.

For the channel information, we have

$$\log P(Y|X) = \log \prod_{i,j} p(y_j^i | x_j^i) = \sum_{i,j} \log p(y_j^i | x_j^i) \tag{2}$$

The traditional Viterbi algorithm indeed maximizes this term. x^i denotes the code-word of sequence X, $x^i = \{x_0^i, x_1^i, \cdots, x_j^i, \cdots\}$, where i and j are the symbol and bit index, respectively; y_j^i is the deteriorated symbol of x_j^i.

For the source information, we have

$$\log P(X) = \log \prod_i p(x^i) = \sum_i \log p(x^i) \tag{3}$$

If only dealing with the case of independent code-words, for each symbol we have

$$\log p(x^i) = \log \prod_j p(x_j^i | x_{k(k<j)}^i) = \sum_j \log p(x_j^i | x_{k(k<j)}^i) \tag{4}$$

The expression $\log P(Y|X) + \log(X)$ is similar to the form of the metric path of a convolution decoder utilizing Viterbi algorithm. The first term $\log P(Y|X)$ depends on the knowledge of channel while the second term relies on the statistics of the source.

2 *A Priori* Bit Probabilities in Huffman Codes

In the decoding of Huffman codes, it may be assumed that the symbol probabilities are known[3,4]. We can show that from these symbol probabilities we can derive *a priori* bit probabilities, which are directly obtained from the tree representation of Huffman codes. To describe the method, we take a code-book as an example. Let $C = \{a, b, c, d\}$ denote the set of all symbols of the Huffman code shown in Fig.1 and Tab.1.

Let each symbol x^i of C be such that $x^i = \{x_0^i, x_1^i, \cdots, x_j^i, \cdots\}$, with i and j representing the symbol and bit index, respectively, let $p(x^i)$ denote its probability, and the probability of symbol c can be written as

$$p(c) = p(x^i = c) = p(x_0^i = 1, x_1^i = 1, x_2^i = 0) \tag{5}$$

i.e. the probability of c is equal to that of the bit sequence $\{1,1,0\}$. Thus we can rewrite it as

$$p(c) = p(x_0^i = 1)p(x_1^i = 1 | x_0^i = 1) \cdot p(x_2^i = 0 | x_0^i = 1, x_1^i = 1) \tag{6}$$

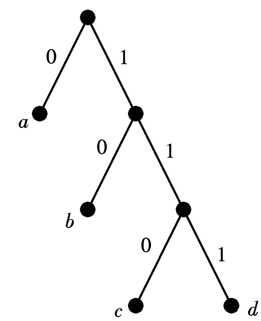

Fig.1 Tree representation of Huffman code

Tab.1 Table of Huffman code

Symbols	a	b	c	d
Code words	0	10	110	111

From the probabilities of code-words we can get the bit transfer probabilities:

$$p(x_0^i = 1) = p(b) + p(c) + p(d) \tag{7}$$

$$p(x_1^i = 1 | x_0^i = 1) = \frac{p(x_1^i = 1, x_0^i = 1)}{p(x_0^i = 1)} = \frac{p(c) + p(d)}{p(b) + p(c) + p(d)} \tag{8}$$

$$p(x_2^i = 0 | x_0^i = 1, x_1^i = 1) = \frac{p(c)}{P(c) + p(d)} \tag{9}$$

We can now generalize these equations to any Huffman code of independent information sequence.

Let x^i denote the Huffman code-word, $x^i = \{x_0^i, x_1^i, \cdots, x_j^i, \cdots\}$ and $x_j^i \in \{0,1\}$. Given bit sequence $x_p^i, p = 0, 1, \cdots, k-1$, the problem is how to compute the probability of x_k^i.

① Search the code-words in which the first k bits(0 to $k-1$) are equal to the first k bits of code-word x^i and denote these code-words with $x^n \in \alpha^k$;

② Find the code-words which the kth bit is equal to $s(s \in \{0,1\})$ from α^k and denote it with $x^m \in \alpha_s^k$;

③ Then we have $p(x_k^i = s | x_{p(p<k)}^i) = \sum p(x^m) / \sum p(x^n), s \in \{0,1\}$.

3 Suggested Algorithm

When the difference of the metrics of the competing paths is large, the decoding according to channel information is reliable, so in this case the source information can be neglected and no change is made. When the difference is small, the decoding algorithm utilizing only the channel information is not reliable enough and so we make use of the source information. Choosing a threshold ε, when the difference is smaller than the threshold, we can decode the receiving sequence as Eq.(1). The decoding principle is shown in Fig.2.

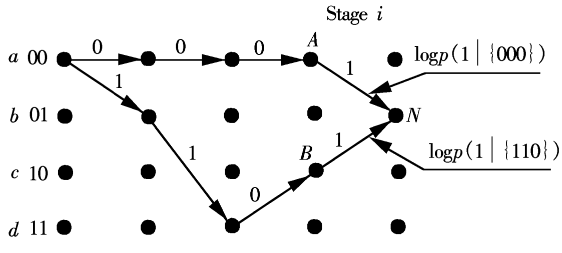

Fig.2 The trellis expression of decoding

Fig.2 illustrates, for the second state in stage i of the trellis, an example of the selection between the sequence of bits $\{0,0,0,1\}$ and $\{1,1,0,1\}$, the *a priori* information for the first sequence is likelihood of receiving bit 1 after getting sequence $\{0,0,0\}$, i.e. $\log(1\{000\})$, while the *a priori* information for the second sequence is one after getting $\{1,1,0\}$, i.e. $\log(1\{110\})$. To use the *a priori* information we should know at which node of the tree representation of the Huffman codes we are. To get this information, we just have to keep in memory the corresponding node of each sequence being decoded. We can now summarize the decoding algorithm as follows.

For each state s_i of the trellis, when the difference of the competing paths' metrics is below threshold ε,

① Compute the *a priori* information of the competing path;

② Add the *a priori* information to the metric branch for both transitions;

③ Preserve the path which has larger metric;

④ Store the corresponding path and metrics;

⑤ Store the node of the tree of the corresponding Huffman code.

The suggested algorithm is to revise the metric path of traditional Viterbi algorithm with the *a priori* information. The *a priori* information can be computed in advance and look up the table when decoding. Adopting the threshold simplifies the decoding procedure when the difference of metrics is large and the slight modification of the metric can be particularly benefit for bad channel conditions. Moreover, the decoding step ⑤ can also reduce the complexity of Huffman decoding since it is no longer required to look up in a table for Huffman decoding.

4 Simulation and Conclusion

The simulation is under AWGN channel. The coefficients after the DCT (discrete cosine transform) of the gray image Lenna 512 × 512 are coded with Huffman codes. The Huffman table makes reference to the JPEG standard. The convolution code generators are $g_1 = (111) = (7)_8, g_2 = (101) = (5)_8$, the bitrate is $R_c = 1/2$ and constraint length is $K = 3$.

Fig.3 is the BER (bit error rate) performance curve of the given channel SNR. We can see that the performance gain is about 1.0 dB when the channel SNR is low, and tends to decrease when the channel condition gets better.

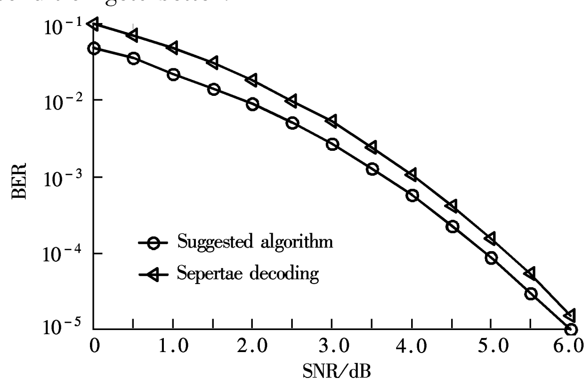

Fig.3 Simulation results

In summary, the suggested algorithm utilizes the *a priori* information of source to improve the decoding performance and simplifies the source decoding.

References

[1] Hagenauer J. Source-controlled channel decoding[J]. *IEEE Trans Communications*, 1995, **43**(9): 2449 - 2457.

[2] Vembu S, Verdu S, Steinberg Y. Source-channel separation theorem revisited[J]. *IEEE Trans Information Theory*, 1995, **41**(1): 44 - 54.

[3] Sayhood K, Out H H, Demir N. Joint source/channel coding for variable length codes[J]. *IEEE Trans Communications*, 2000, **48**(5): 787 - 794.

[4] Park M, Miller D J. Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation[J]. *IEEE Trans Communications*, 2000, **48**(1): 1 - 6.

与 Huffman 码相结合的卷积码软判决译码方案

郭东亮　陈小蔷　吴乐南

（东南大学无线电工程系，南京 210096）

摘　要　提出了一种与 Huffman 码相结合的卷积码软判决译码方案.对卷积码的软判决维特比译码算法进行了改进,由 Huffman 编码的码字概率计算出比特转移概率,进而得出与维特比译码的支路似然值相对应的信源先验信息,通信系统的编码端不作改动,当由于信道条件恶化等原因造成维特比译码算法的支路量度相差很小而难以进行可靠译码时,将信源先验信息作为支路量度的修正值,以改善译码的性能.与分离的信源、信道译码相比,性能增益约为 0.5～1.0 dB,增加的复杂性很小.仿真实验验证了算法的有效性.

关键词　软判决维特比译码，先验信息，Huffman 编码,卷积码

中图分类号　TN911.21