Iterative Source-Channel Decoding for Robust Image Transmission

Wei Xiang, Steven S. Pietrobon and S. Adrian Barbulescu

***Abstract*— In this paper we investigate iterative source-channel decoding for robust JPEG coded images transmission over noisy channels. Huffman codes used as the variable-length coding scheme in JPEG coding can be represented by an irregular trellis structure proposed by Balakirsky (1997). State transition probabilities can be derived from the irregular trellis and can be used as *a priori* information to help iterative decoding between channel and source *a posteriori* probability (APP) decoders. Iterative decoding of JPEG coded images does not perform well due to the poor distance property of the original JPEG Huffman codes. In this paper, we proposed a symmetric reversible variable length code with free distance $d_f = 2$ which can dramatically improve the system performance when iterative decoding is adopted. A maximum coding gain of 4 dB was observed from the simulation results. Subjective results in terms of reconstructed images were also presented.**

***Index Terms*— Iterative source-channel decoding, Reversible variable length code, JPEG coded image.**

I. Introduction

There has been increasing demand on multimedia services over fixed and wireless channels over the last decade. Robust image or video transmission systems are critical to such multimedia services. Shannon's classical separation theorem states that we can optimise the end-to-end system design by separately optimising the source coders and the channel coder [1].

For the past decades, Shannon's separation principle has stood for a justification for separate source and channel coding. However, this result holds true only for infinite source code dimension and infinite long channel code. Significant progress has been made throughout last decades to optimise each individual module of communication systems. The innovative next generation of mobile network will be globally integrated architecture where individual modules are jointly designed to enable simultaneous optimisation of bandwidth as well as Quality of Service (QoS). Since ideal hypotheses of separate source-channel coding (SSCC) put unrealistic constrains on the system, a joint source-channel coding (JSCC) design may reduce complexity and delay to yield better end-to-end system performance.

Previous JSCC research work has concentrated on an optimal joint source-channel rate allocation strategy, which was computationally expensive to be adopted in practical systems. However, these methods do not perform well enough for low channel signal-to-noise ratios (SNRs), although they provide excellent results for moderately distorted channels.

On the other hand, joint source-channel decoding (JSCD) [2] is another approach to combat channel noise by exploiting the residual redundancy of the source encoder at the receiver side. The source redundancy can be utilised as *a priori* information of the source statistics to help decoding.

To further exploit the "turbo" principle [3] to joint source-channel decoding, Bauer and Hagenauer proposed an iterative source-channel decoding approach for variable lengh coded sources [4] [5]. By modifying the variable length source decoder to an *a posteriori* probability (APP) decoder, the approach allows soft extrinsic information exchange between the channel APP decoder and the source VLC APP decoder. Their approach yields superior results compared to separate source and channel decoding. The fact that variable length codes can be represented with an irregular trellis structure [6] makes iterative source-channel decoding possible.

In many international image/video standards like JPEG [7] and MPEG4 [8], variable-length codes (VLCs) such as Huffman codes are used as the entropy compression scheme. Due to the error propagation property of VLCs, the variable length coded signals are very prone to channel noise. Even a single bit error could destroy the whole coded bit-stream if there are no other error resilient schemes adopted. Inspired by Bauer and Hagenauer's work, we applied iterative source-channel decoding to JPEG coded images with channel coding. In [9], *a priori* bit probabilities are derived from the tree representation of Huffman codes and incorporated to help turbo decoding. In this paper, we show how the state transition probability (STP) can be derived from the trellis representation of VLCs. We develop a general formula to compute STPs for a given VLC-trellis. The STP serves as a built-in source of *a priori* information to help the iterative source-channel receiver.

In this paper, we studied iterative source-channel decoding applied to JPEG coded images. We start with Section II to derive the state transition probabilities based on the VLC-trellis of Huffman codes. In Section III, we formulise the APP decoding of Huffman codes and show how the STP can be incorporated into the MAP algorithm. Section IV discusses the issues of designing RVLCs with large free distance which is critical for iterative decoding. In Section V, iterative decoding of JPEG coded images with channel coding is addressed. Section VI presents simulation results. Finally, Section VII is dedicated to conclusions and future work.

The authors are with the Institute for Telecommunications Research, University of South Australia, SA 5095, Australia, e-mail:{wei, adrian}@spri.levels.unisa.edu.au. Steven S. Pietrobon is also with Small World Communications, 6 First Avenue, Payneham South SA 5070, Australia, e-mail:{steven}@sworld.com.au.

This work was carried out in part with support from the Commonwealth of Australia through the Cooperative Research Centre for Satellite Systems.

II. State Transition Probabilities of VLC-Trellis

A variable length coded sequence can be regarded as a sequence of bits instead of concatenated variable length codewords. Thus, the variable length coded sequence can be represented by a simple but effective trellis structure proposed in [6]. As shown in [5], a bit-level soft-in/soft-out (SISO) decoding module for VLCs was derived by applying the BCJR algorithm [10] to such a VLC-trellis. Therefore, an APP VLC source decoder can be implemented to generate soft outputs and exchange extrinsic information between the source VLC APP decoder and channel APP decoder.

For a variable length coded sequence, the symbol or codeword probabilities are usually assumed to be known. In [9], *a priori* bit probabilities were derived directly from the tree representation of Huffman codes using the symbol probabilities of the codes. The *a priori* bit probabilities were used in a joint source-channel decoding process. In contrast to their approach, we proposed a method to derive the state transition probabilities (STP) based on the bit-level trellis representation of variable length codes. In the next section, we show how the STP can be incorporated into the maxium *a posteriori* (MAP) algorithm to improve system performance.

As a comparison, we chose as an example the Huffman code in [9] to demonstrate how to derive the state transition probabilities from the trellis. The variable length code $\mathcal{C} = \{00, 01, 10, 110, 111\}$ and the corresponding codeword probabilities are $\mathcal{P} = \{p_1, p_2, p_3, p_4, p_5\}$.

Fig. 1 shows the tree representation and the bit-level trellis of the variable length code $\mathcal{C}$. The nodes in the tree are subdivided into a root-node (R), internal nodes (I) and terminal nodes (T). We can treat the VLC-trellis as an irregular trellis in comparison to the trellis of a recursive systematic convolutional (RSC) code. The trellis of an RSC code has nice systematic properties. Each state has two paths emanating and two paths merging with the same probabilities. In contrast to the systematic trellis of an RSC code, each state in the VLC-trellis might have an arbitrary number of emanating or merging paths or even have parallel paths emanating and merging between states. Moreover, different paths might emanate or merge to a state with different probabilities from each other because the codeword probabilities of VLCs are in general different. We can utilise the codeword probabilities of VLCs to derive the state transition probabilities which are used as *a priori* information to facilitate iterative source-channel decoding.

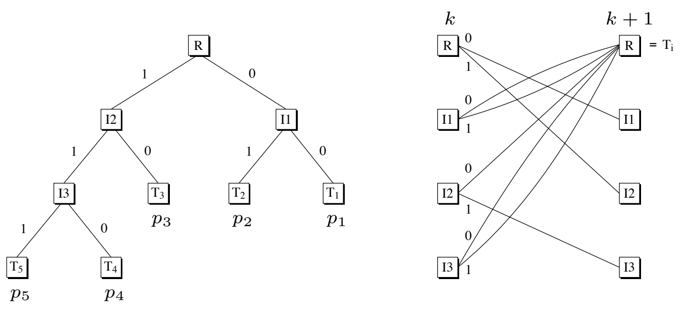

Fig. 1. Tree representation and VLC-trellis with parallel transitions.

Let $\mathcal{P}$ be the probability, S the state, k the time instance and d_k the input bit of a VLC-trellis. For an arbitary irregular trellis, we define the state transition probabilities between two adjacent states as $\mathcal{P}(S_{k+1} = n|S_k = m)$. For instance in Fig. 1, the STP from internal node I2 to the root node R can be written as

$$\mathcal{P}(S_{k+1} = \mathrm{R}|S_k = \mathrm{I2}) = \frac{p_3}{p_3 + p_4 + p_5}. \quad (1)$$

The STP from internal node I2 to the I3 can be written as

$$\mathcal{P}(\mathcal{S}_{k+1} = \mathrm{I3}|\mathcal{S}_k = \mathrm{I2}) = \frac{p_4 + p_5}{p_3 + p_4 + p_5}. \quad (2)$$

As we can see, there are two paths emanating from the I2 node, merging to the R node and the I3 node, respectively. In contrast to a regular RSC trellis in which each path has equal probability, the two paths have different probabilities. This is because variable length codes are constructed in such a way that the more frequent codewords have shorter code length. Therefore, some paths have priority over the other for an irregular VLC-trellis. The STP contains this priority information which can be used as *a priori* information by the source APP decoder.

Note that there are parallel transition paths which merge at the same state if the different codewords of a variable length code have the same length. In the case of parallel transition paths, the two paths are distinguished by the input bit i. Thus the STP can be associated with an input bit d_k and becomes $\mathcal{P}(S_{k+1} = n, d_k = i|S_k = m)$. The principle of calculating the STP still applies. For instance in Fig. 1, there are two parallel paths from internal node I1 to the root node R. They are distinguished by an input bit 0 or 1. The STP from I1 to R associated with the input bit 0 is

$$\mathcal{P}(S_{k+1} = \mathrm{R}, d_k = 0|S_k = \mathrm{I1}) = \frac{p_1}{p_1 + p_2}. \quad (3)$$

The STP from internal node I1 to the R associated with the input bit 1 is

$$\mathcal{P}(\mathcal{S}_{k+1} = \mathrm{R}, d_k = 1|\mathcal{S}_k = \mathrm{I1}) = \frac{p_2}{p_1 + p_2}. \quad (4)$$

For a generic trellis, we formulise the state transition probabilities between two adjacent states associated with an input bit i as

$$\mathcal{P}(S_{k+1} = n, d_k = i|S_k = m) = \frac{\sum_{\alpha \in f(S_{k+1}, i)} p_\alpha}{\sum_{\beta \in g(S_k)} p_\beta}, \quad (5)$$

where $f(S_k, i)$ are all the forward nodes indices connected to S_k with the input bit i and $g(S_k)$ are all the forward nodes indices connected to S_k. For example, I1 is connected to R with input bit 0, so $f(\mathrm{R}, 0) = \{1\}$. T_3, T_4 and T_5 are connected to I2, so $g(\mathrm{I2}) = \{3, 4, 5\}$.

The major advantage of using the STP is that it can be naturally incorporated into the MAP algorithm. There is no need to run a second algorithm in parallel to keep node and state information at each time instance of the VLC-trellis, as is the case in [9].

III. APP DECODING OF VLCS

For the purpose of iterative source-channel decoding, we need a source decoder that produces soft outputs. The soft output Viterbi algorithm or the BCJR algorithm can be applied to the VLC-trellis to decode variable length codes. In [9] a modified Viterbi algorithm was derived. The *a priori* probabilities of Huffman codes were used to improve the performance of the Viterbi algorithm. The variable length coded sequence can be treated as a sequence of bits instead of a concatenation of codewords [5]. In [4], bit-MAP decoding was developed to compute the *a posteriori* probabilities of bits of the variable length coded sequence.

Similar to the algorithm in [4] and following the conventions in [11], we modified the BCJR algorithm for the soft decoding of the VLC-trellis.

Denote by d_k and R_k the data bit and received symbol at time k. We define the log-likelihood ratio (LLR), λ_k associated with each decoded bit d_k as

$$\begin{aligned}\lambda_k &= \log\frac{Pr(d_k=0|R_1^N)}{Pr(d_k=1|R_1^N)} \\ &= \log\frac{\sum_m\sum_{m'}\lambda_k^{0,m,m'}}{\sum_m\sum_{m'}\lambda_k^{1,m,m'}}, \end{aligned} \quad (6)$$

where $Pr(d_k=i|R_1^N)$ is the *a posteriori* probability (APP) of the data bit d_k.

The joint probability in (6) is defined as follows

$$\begin{aligned}\lambda_k^{i,m,m'} &= Pr(d_k=i,S_k=m,S_{k+1}=m',R_1^N) \\ &= \underbrace{Pr(R_1^{k-1},S_k=m)}_{\alpha}\underbrace{Pr(R_{k+1}^N|S_{k+1}=m')}_{\beta} \\ &\quad \underbrace{Pr(d_k=i,S_{k+1}=m',R_k|S_k=m)}_{\delta} \end{aligned} \quad (7)$$

The foward state metric α_k^m can be recursively caluculated as follows:

$$\begin{aligned}\alpha_k^m &= Pr(R_1^{k-1}|S_k=m)Pr(S_k=m) \\ &= \sum_{m'}\sum_{i=0}^{1}\alpha_{k-1}^{m'}\delta_{k-1}^{i,m',m}. \end{aligned} \quad (8)$$

Similarly we can derive the recursive reserve state metric β_k^m as follows

$$\begin{aligned}\beta_k^m &= Pr(R_k^N|S_k=m) \\ &= \sum_{m'}\sum_{i=0}^{1}\beta_{k+1}^{m'}\delta_k^{i,m,m'}. \end{aligned} \quad (9)$$

Using Bayes' rule, the branch metric $\delta_k^{i,m,m'}$ can be expressed as

$$\begin{aligned}\delta_k^{i,m,m'} &= Pr(d_k=i,S_{k+1}=m',r_k|S_k=m) \\ &= Pr(R_k|d_k=i)Pr(S_{k+1}=m',d_k=i|S_k=m). \end{aligned} \quad (10)$$

For an AWGN channel with zero mean and variance σ^2, (10) becomes

$$\delta_k^{i,m,m'} = \chi_k\exp(L_c r_k i)Pr(S_{k+1}=m',d_k=i|S_k=m), \quad (11)$$

where χ_k is a constant and $L_c=2/\sigma^2$. The last term in (11) is the STP which can be obtained from the VLC-trellis from (5). Note that the recursive calculation of both the forward state metric α_k^m and reverse state metric β_k^m use the same STP.

IV. REVERSIBLE VLC CONSTRUCTION FOR JPEG HUFFMAN CODES

The JPEG [7] still image compression standard has proved to be a success story in source coding techniques. The JPEG standard consists of three core components, *i.e.,* two-dimension discrete cosine transform (2D-DCT), scalar quantisation and variable length entropy coding. In this study, Huffman codes [12] are used as the entropy coding scheme.

We consider JPEG coded image communication systems with rate 1/2 channel coding. The conventional "separate" communication system is shown in Fig. 2. The transmitter consists of a JPEG source encoder and a channel encoder. The receiver performs the separate source and channel decoding, *i.e.,* no information exchange between the JPEG source decoder and the channel decoder. Previous work has revealed that unequal error protection (UEP) can improve the system performance [13].

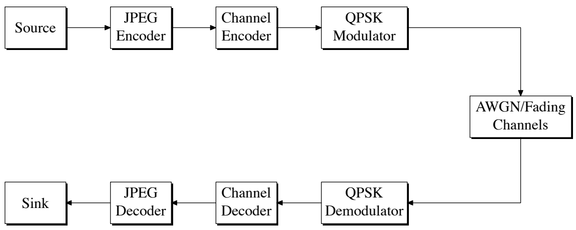

Fig. 2. Conventional JPEG communication system

Denote a sequence of M-ary i.i.d. source symbols by $\mathbf{u}=(u_1,u_2,\ldots,u_M)$ with probability density function $\mathcal{P}=(p_1,p_2,\ldots,p_M)$. Let a variable length code $\mathcal{C}$ map a sequence $\mathbf{u}$ of M symbols into a binary sequence $\mathbf{b}=(b_1,b_2,\ldots,b_N)$ of length N. Consider the whole sequence $\mathbf{b}$ as one particular codeword of a code $\mathcal{B}$, whose codewords are all possible combinations of codewords $c(u)$ of the VLC $\mathcal{C}$ with total length N. The error-correcting capability of a VLC is dominated by the free distance metric d_f which is defined in [5] as

$$d_f = \min\{d_H(b^{(i)},b^{(j)})|b^{(i)},b^{(j)}\in\mathcal{B};b^{(i)}\neq b^{(j)}\} \quad (12)$$

where $d_H(b^{(i)},b^{(j)})$ is the Hamming distance between two codewords $b^{(i)}$ and $b^{(j)}$ in $\mathcal{B}$.

Due to the nonlinearity of code $\mathcal{B}$, direct evaluation of d_f is relatively complicated. However, one simpler VLC distance metric called minimum block distance d_b can be utilised to reduce the evaluation complexity of free distance d_f [14]. The minimum block distance d_b is define in [15] as the minimum

Hamming distance between two codewords with equal length of $\mathcal{C}$

$$d_b = \min\{d_H(c(u_i), c(u_j))|c(u_i), c(u_j) \in \mathcal{C}; \quad l(u_i) = l(u_j); u_i \neq u_j\}. \tag{13}$$

As derived in [14], the relationship between d_b and d_f for Huffman codes and RVLs is

$$d_f \geq \min(2, d_b). \tag{14}$$

Relationship (14) can be understood in a way that all RVLCs which do not have equal-length codewords have $d_f \geq 2$. Moreover, for an RVLC containing codewords of equal length, $d_b \geq 2$ is a sufficient condition for $d_f \geq 2$.

As far as soft decoding of VLCs is concerned, RVLCs with large d_f usually have better performance in terms of symbol error rate. Therefore, free distance d_f is a critical design criterion one has to take into consideration to construct an RVLC from an existing Huffman code. In this paper, we consider constructing RVLCs for JPEG Huffman codes.

For a gray JPEG coded image, there are two Huffman code tables. There is a 12-entry Huffman table for the DC luminance component, while there is a 162-entry Huffman table for the AC luminance component [7]. As will be showed in Section VI, the original JPEG Huffman codes do not give good performance since they have $d_f = 1$. Due the their poor distance property, the convergence behaviour of JPEG Huffman codes, which is important for the iterative source-channel decoding, are bad. Using the criterion provided by (14), we can construct RVLs with $d_f = 2$ from existing JPEG Huffman codes. In Table I, we constructed a symmetric RVLC $\mathcal{C}_R$ for the JPEG Huffman code $\mathcal{C}_H$ for the luminance DC component. Note the probability distribution $\mathcal{P}$ of DC symbols in Table I was obtained by training on "Lena" and "Goldhill" images. Code $\mathcal{C}_R$ has $d_f = 2$ at the expense of slightly increased average length. Moreover, code $\mathcal{C}_R$ is symmetric. This means that it has the nice property of needing only one code table for both forward and backward decoding which is an advantage from the viewpoint of memory usage and simplicity [16].

TABLE I

ORIGINAL JPEG HUFFMAN CODE FOR LUMINANCE DC COMPONENT VERSUS A PROPOSED SYMMETRIC RLVC WITH $d_f = 2$

DC Symbol	Occurrence Probability $\mathcal{P}$	JPEG Lumin. DC Huffman Code $\mathcal{C}_H$	Sym. RVLC $\mathcal{C}_R$ with $d_f = 2$
0	0.371745	00	00
1	0.071615	010	111
2	0.102214	011	010
3	0.147135	100	101
4	0.132812	101	0110
5	0.124349	110	1001
6	0.049479	1110	11011
7	0.000651	11110	01110
8	0	111110	10001
9	0	1111110	011110
10	0	11111110	100001
11	0	111111110	110011
Average codeword length		2.6790	2.9857

V. ITERATIVE DECODING OF JPEG CODED IMAGES

In this paper we examine iterative decoding of JPEG coded image communication systems. The receiver implements iterative decoding between the Huffman APP decoder and the channel decoder. Two transmission schemes were studied. The first scheme is a conventional serial concatenated system. The transmitter consists of a JPEG source encoder and an RSC channel encoder as illustrated in Fig. 3. They are separated by a pseudo-random interleaver. We denote the image pixel vector to the JPEG encoder by $\mathbf{u} = (u_1, u_2, \ldots, u_M)$, the bit-stream out of the JPEG encoder by $\mathbf{b} = (b_1, b_2, \ldots, b_N)$, the information bit sequence and parity bit sequence out of the rate $1/2$ RSC channel encoder by $\mathbf{x} = (x_1, x_2, \ldots, x_K)$ and $\mathbf{p} = (p_1, p_2, \ldots, p_K)$, respectively.

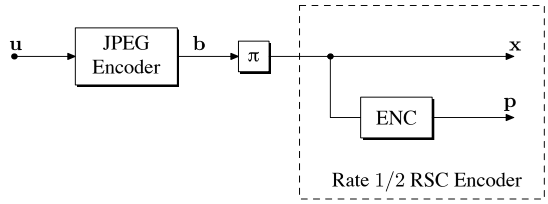

Fig. 3. Transmitter of a serial concatenated system.

The coded bit-stream is sent through an additive white Gaussian noise (AWGN) or Rayleigh fading channel. On the receiver side, we denote the received noisy version of $\mathbf{x}$ and $\mathbf{p}$ by $\mathbf{X}$ and $\mathbf{P}$, respectively. The iterative receiver of the serial concatenated system is depicted in Fig. 4. It consists of an inner channel APP decoder and an outer Huffman APP decoder.

At the first decoding iteration, the input to the channel APP decoder is the soft output of the noisy channel $L_c\mathbf{X}$. The channel decoder computes the APP $P(\mathbf{x}|\mathbf{X})$ of the information bits $\mathbf{x}$. We denote the extrinsic output of the channel and source APP decoders by Z_c and Z_s. The log-likelihood ratio of the inner channel decoder is $L^{(C)}(\mathbf{x}) = L_c\mathbf{X} + Z_s + Z_c$, where Z_s is the *a priori* information from the extrinsic output of the soft Huffman decoder which is set to 0 at the initial iteration. Z_c refers to the extrinsic output of the channel APP decoder. Subtracting the *a priori* term from $L^{(C)}(\mathbf{X})$, $L_c\mathbf{X} + Z_c$ are fed into the outer Huffman soft decoder. The soft output of the Huffman APP decoder is $L^{(S)}(\mathbf{b}) = L_c\mathbf{b} + Z_c + Z_s$. Subtracting the soft input from the soft output for the outer source decoder, the extrinsic information Z_s of the Huffman APP decoder is passed to the channel APP decoder as *a priori* information. At the final iteration, the hard output of the iterative receiver is switched to the JPEG decoder to perform normal JPEG decoding. $\mathbf{U}$ is the estimated output of the image pixel vector $\mathbf{u}$.

VI. SIMULATION RESULTS

In this section we present our simulation results on the performance of both the original Huffman DC code $\mathcal{C}_H$ and the proposed symmetric RVLC $\mathcal{C}_R$. We transmitted 1024 differential pulse code modulation (DPCM) coded DC symbols from "Lena" 256 × 256 image over the AWGN channel. The coded bitstream out of the JPEG DC DPCM encoder was protected by a 16-state inner code with code polynomials 35/23 in octal notation. The channel coding rate for the coded bitstream using

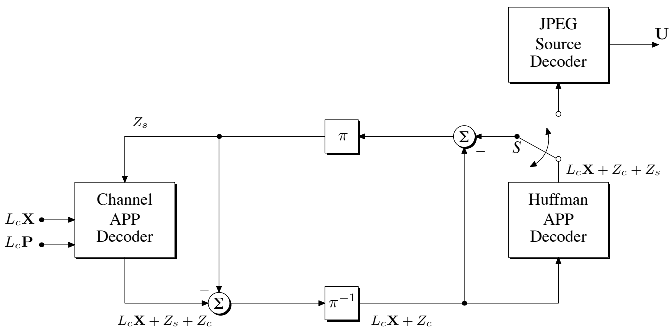

Fig. 4. Iterative receiver of serial concatenated system.

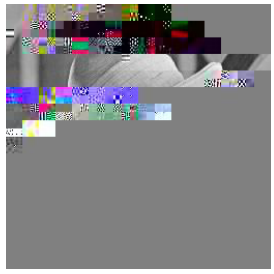

Fig. 6. Reconstructed image using $\mathcal{C}_H$, $E_b/N_0 = 1.5$ dB

Fig. 7. Reconstructed image using $\mathcal{C}_R$, $E_b/N_0 = 1.5$ dB

Huffman code $\mathcal{C}_H$ was 1/2. Due to the slightly increased average length of RVLC $\mathcal{C}_R$, we punctured the coded bitstream so that the overall coding rate was also 1/2 to ensure fair comparison.

Objective simulation results are presented in Fig. 5. Note the symbol error rate (SER) is measured in terms of the Levenshtein distance which is more appropriate than the measurement of a simple symbol comparison [5]. As can be seen, iterative decoding using the proposed symmetric RVLC $\mathcal{C}_R$ remarkably outperforms the one using the original JPEG Huffman code $\mathcal{C}_H$. A maximum coding gain of 4 dB is obtained.

Subjective results with "Lena" 256 × 256 image were presented in Figs. 6 and 7. We transmitted the target image over the noisy AWGN channel. Please note that only DC part is affected by the noise but other parts of the image are error-free since we are mainly interested in the comparison of $\mathcal{C}_H$ and $\mathcal{C}_R$ in terms of iterative decoding in this study. Figure 6 showed the reconstructed image using Huffman code $\mathcal{C}_H$ at E_b/N_0 of 1.5 dB. Figure 7 presented the reconstructed image using RVLC $\mathcal{C}_R$ at E_b/N_0 of 1.5 dB. Significant visual difference can be perceived from the comparison of Fig. 6 and Fig. 7.

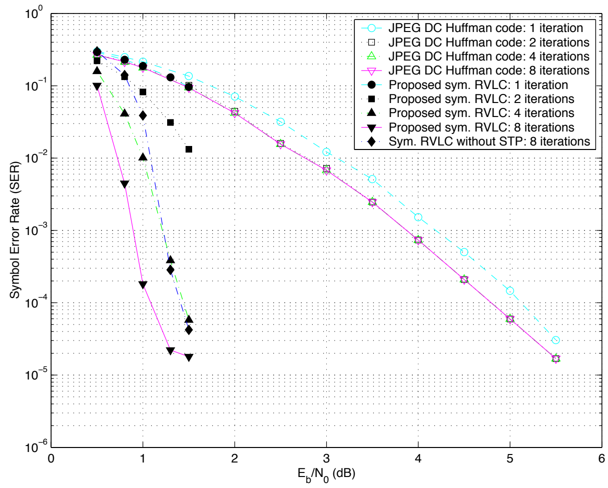

Fig. 5. Simulation results for iterative decoding using Huffman code $\mathcal{C}_H$ and proposed symmetric RVLC $\mathcal{C}_R$.

VII. Conclusions

In this paper we studied iterative decoding of JPEG coded images with channel coding. Representing the VLCs with a bit-level trellis with possible parallel transitions, we showed the derivation of the state transition probabilities. The STPs can be used as *a priori* information to improve the APP decoding of VLCs.

We proposed a symmetric RVLC with $d_f = 2$ for the JPEG luminance DC component. Compared to the original JPEG Huffman code $\mathcal{C}_H$, the proposed RVLC $\mathcal{C}_R$ has a significant performance improvement when iterative decoding is performed.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October 1948.
[2] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, pp. 2449–2457, Sept. 1995.
[3] J. Hagenauer, "The Turbo principle: Tutorial introduction and state of the art," in *Proceedings of the International Symposium on Turbo Codes*, Brest, France, pp. 1–11, September 1997.
[4] R. Bauer and J. Hagenauer, ""Turbo-FEC/VLC-Decoding" and its application to text compression," in *International Conference on Information Sciences and Systems*, Princeton University, March 2000.
[5] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel decoding," in *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, USA, pp. 272–282, March 2001.
[6] V. B. Balakirsky, "Joint source-channel coding with variable length codes," in *Proceedings of IEEE ISIT*, Ulm, Germany, 1997.
[7] ISO/IEC JTC1, *Digital compression and coding of continuous-tone stillimages Part 1: requirements and guidelines*. JPEG-9-R6 CD10918-1, 1991.
[8] ISO/IEC JTC1/SC29/WG11, *MPEG-4 video verification model version 11.0: Coding of moving pictures and associated audio information*. Touradj Ebrahimi, 1998.
[9] L. Guivarch, J. Carlach, and P. Siohan, "Joint source-channel soft decoding of Huffman codes with turbo-codes," in *Proceedings of DCC*, Snowbird, UT, 2000.
[10] L. R. Bahal, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, pp. 284–287, March 1974.
[11] S. S. Pietrobon, "Implementation and performance of a Turbo/MAP decoder," *Int. J. of Satellite Commun.*, vol. 16, pp. 23–46, Jan.-Feb. 1998.
[12] D. A. Huffman, "A method for the construction of minimum redundancy codes," in *Proceedings of the IRE*, vol. 40, pp. 1098–1101, 1951.
[13] W. Xiang, S. A. Barbulescu, and S. S. Pietrobon, "Unequal error protection applied to JPEG image transmission using turbo codes," in *Proc. IEEE Information Theory Workshop (ITW'2001)*, Cairns, Australia, pp. 64–66, Sept. 2001.
[14] K. Lakovic and J. Villasenor, "On design of error-correcting reversible variable length codes," *IEEE Communications Letters*, vol. 6, pp. 337–339, August 2002.
[15] V. Buttigieg and P. G. Farrell, "A maximum likelihood decoding algorithm for variable-length error-correcting codes," in *5th Bangor Sym. on Communications*, Bangor, U.K., pp. 56–59, 1993.
[16] Y. Takishima, M. Wada, and H.Murakami, "Reversible variable length codes," *IEEE Transactions on Communications*, vol. 43, pp. 158–162, Feb.–Apr. 1995.