Facsimile Image Compression using Slepian-Wolf Coding Principles and Context Modelling

Raghunadh K. Bhattar, K.R.Ramakrishnan, and K.S.Dasgupta

***Abstract*—In this paper, Slepian-Wolf coding is considered for compression of facsimile images. The correlation between the image scan lines is exploited at the decoder using Slepian-Wolf coding (SWC). The side information is generated at the decoder using previously decoded image lines and context modelling. A novel weighted context-based iterative decoding of low density parity check (LDPC) codes is proposed in addition to realizing context-based initialization of log-likelihood ratio values in LDPC iterative decoding. On an average, an increase of 0.60 in the compression ratio is obtained with the context modelling. As the proposed method is based on SWC principles, it is inherently error resilient and affords a system with a low-complexity encoder.**

***Index Terms* — Facsimile, LDPC, Context-Modelling, Nonuniform Source, Slepian-Wolf Coding, DSC.**

I. Introduction

The Slepian-Wolf (SW) theorem [1] is generally applied to uniform binary sources to exploit the correlation between two binary sources. Slepian-Wolf coding is also known as distributed source coding (DSC), or compression of correlated sources with side information. DSC schemes offer lot of scope for a flexible trade-off between encoder and decoder complexities. The basic theory and principles of DSC can be found in [2]. DSC principles are applied to remote sensing images and multi-view images to realize a low-complexity encoder by exploiting inter-band correlation and correlation between the images respectively. However, literature on a single-image compression utilizing spatial redundancy in accordance with DSC principles is limited. In [3], a still image was divided into two sources, and both the sources were lossy coded. One source was encoded using a low-pass component of wavelet decomposition and the other source was encoded using a modulo based binning scheme in tune with the DSC principles. The authors of [4] considered single-source compression within the DSC framework using 'virtually' created side information. In [5], the authors constructed distributed image compression techniques that operate in the pixel domain and exploit the correlation only at the decoder for binary text images.

Recently, wireless facsimiles [6] and portable facsimiles have been gaining popularity for different applications. These devices need to consume less power for longer battery life.Generally, lossless compression is desirable for facsimile (or text) images. Typically, facsimile images contain significant redundancy, and they can be treated as non-uniform sources with highly biased source distribution (around $P(X = 0) = p0 = 0.96$ [7]). A direct application of Slepian-Wolf coding techniques has been investigated in this paper. These techniques are based on the compression of a non-uniform source under the DSC paradigm. Recently, LDPC codes [8] were effectively employed in [9], [10] for the compression of nonuniform sources. The authors showed that LDPC codes are better suited for non-uniform sources and can come close to achieving the Slepian-Wolf bound. Hence, in this paper, we explore the use of LDPC codes for the compression of facsimile images under the DSC paradigm.

The Consultative Committee on International Telephone and Telegraph (CCITT) of the International Telecommunications Union (ITU) standardized several standards for facsimile compression. Notable standards include CCITT group 3 and group 4, more commonly known as fax3 and fax4 compression standards, respectively. An excellent review of facsimile compression methods and standards can be found in [11], [12].

Binary image compression standards, JBIG [13] and JBIG2 [14] standards achieve higher compression ratios using context-based arithmetic coding. On the other hand, context modelling is rarely used in DSC. According to literature, context modelling in DSC was first introduced in [15], where authors reported an overall rate saving of 36%. Here, the context information is derived from the bit planes; the theory underlying this mechanism is somewhat similar to JPEG-2000 context modelling. One drawback of context-based arithmetic coders such as those used in JBIG and JBIG2 is that they are generally very sensitive to channel errors. However, context modelling in DSC is entirely performed at the decoder, and hence, it is less susceptible to channel noise.

In this paper, we have combined DSC and context modeling at the LDPC decoder. The side information at the LDPC decoder is also part of the compressed bit stream. Further, we have introduced a novel context-based LDPC iterative decoding in addition to realizing context-based initialization of log-likelihood ratio (LLR) values in LDPC decoding. As the proposed method is based on DSC principles, it is inherently error resilient and affords a system with a low-complexity encoder.

This paper is organized as follows. Section II describes the different building blocks used in the proposed system. The decoding algorithm used for context-based decoding is presented in Section III. Details of the simulation

Manuscript received April 10, 2011; revised September 14, 2011.

Raghunadh K Bhattar is with EE Dept-IISc, Bangalore and SNAA-SACISRO, Ahmedabad. India (email: k_raghunadh@rediffmail.com)

Prof. K.R.Ramakrishnan is with EE Dept-IISc, Bangalore, India. (krr@ee.iisc.ernet.in)

Prof. K.S.Dasgupta is with IIST Thiruvananthapuram, India (ksdasgupta@yahoo.com) and was formaly with SNPA/SAC/ISRO, Ahmedabad, India.

experiments conducted and test results are reported in Section IV. Finally, Section V presents the conclusion of this paper.

II. Building Blocks of Encoding

DSC principles are combined with classical facsimile image coding techniques to achieve low-complexity facsimile image encoding. The different building blocks used in the proposed scheme are presented in this section. The block diagram of our scheme is shown in Fig. 1.

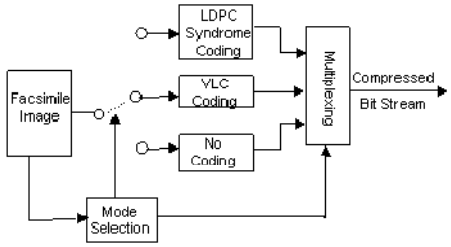


Fig. 1. Block Diagram of Facsimile Image Compression

A. Variable Rate LDPC Codes

Each line in an image has a different source distribution, and consequently, a different compression ratio. Hence, one needs to select a matching LDPC code for each line. This necessitates the use of variable rate LDPC codes. To use the correct LDPC code for each image line in order to achieve lossless decoding, we need to know the simulation threshold values ($q^{\#}$) [9] of each code. For this purpose, a set of LDPC codes of different code rates are pre-designed, and their threshold values are computed.

Generally, each line in the image is encoded independently using LDPC syndrome codes according to (1).

$$s = Hx \tag{1}$$

where H is a $m \times n$ LDPC matrix, x is the input vector (i.e. image line) of length n, and s is the syndrome vector of length m.

B. Variable Length Codes

It may not always be beneficial to code each line with an LDPC code because it may not always be possible to find a suitable LDPC code from the set of pre-designed codes. This may happen when the image lines are very sparsely distributed, very uniformly distributed, or when they do not have any correlation with the previous line or lines. In these situations, a variable length code may be used. Hence, we propose to use suitable VLC (may be in conjunction with run-length coding) instead of LDPC codes for some image lines.

C. No Coding

In binary images, such as facsimile and text, the image line to be coded may often has the same data as the previous line. In these cases, there is no need to encode these lines. The fact that the status of this line is the same as that of the previous line is stored as header information and communicated to the decoder.

D. Mode Selection

A mode selection module selects the type of coding to be applied to the i^{th} line (present line) on the basis of the $(i\text{-}1)^{th}$ line (previous line). Normally, LDPC syndrome coding is used on the image lines. For certain images lines, other modes such as 'VLC coding' or 'No coding' are selected by this module to achieve maximum compression with lossless reconstruction. In a simple decoder, the previous line data is used as side information, and the present line is decoded according to SW principles. When context modelling is used in SW decoding, the scenario becomes more complicated; this will be discussed in detail in later sections.

Assuming that the previous line is input to the virtual binary symmetric channel for SW decoding, we calculate the crossover probability (q) for the present line (i.e. the i^{th} line) as follows.

$$q = \frac{1}{N}\sum_{j=1}^{N} \mathbf{A}(i,j) \oplus \mathbf{A}(i-1,j) \tag{2}$$

where $\oplus$ indicates exclusive-OR operation and A is an M£ N facsimile image. If $q = 0$, i.e. if the data of the present line is the same as that of the previous line, then there is no data to be coded, and this information is embedded as header information. If $q \neq 0$, then a suitable LDPC code is selected such that the $q \leq q^{\#}$, and the syndrome length is minimum for this code among the codes of the given LDPC code set corresponding to a given source distribution p_0 for the i^{th} line. p_0 is given by

$$p_0 = 1 - \frac{1}{N}\sum_{j=1}^{N} \mathbf{A}(i,j) \tag{3}$$

The VLC coded data bits for the i^{th} line are estimated and compared with the syndrome length of the selected LDPC code. If the syndrome length is less than the VLC estimation, then the line is coded with the LDPC syndrome, else the line is coded with VLC codes. If there is no suitable LDPC code (which implies that $q \leq q^{\#}$ is not satisfied for any LDPC code in the given set) and the VLC code is longer than the image line, then the line is transmitted as raw data.

As described above, the encoding mode is selected on the basis of the virtual crossover probability (q) between the present line and the previous line and the threshold values ($q^{\#}$) of the pre-designed LDPC codes. In evaluating the threshold values, the source distribution p_0 is fixed, and the threshold values are estimated for the pre-designed LDPC codes. The threshold values change with p_0; higher the p_0, higher is the threshold value. Obviously, it is not possible to estimate the threshold values (using Monte Carlo methods) for every p_0 as p_0 is a continuous variable. To avoid computing the threshold values for every p_0, we compute the threshold values for one particular source distribution, say $p_0 = 0.9$ (one can choose any other value for p_0 as well) and derive the threshold values for other source distributions. This is possible by exploiting a favourable property reported in [9], according to which the conditional entropy H(X/Y) of source X with side information Y is constant for any source distribution when the LDPC codes are used optimally i.e. at the threshold values of the codes. H(X/Y) is given by

$$H(X/Y) = H(p_x) + H(q^{\#}) - H(p_y)$$

and

$$p_y = p_x(1-q^{\#}) + (1-p_x)q^{\#}$$

where H(.) is the binary entropy function and p_x and p_y are the source distributions of X and Y respectively. The threshold values for source distribution $p_x = 0.9$ are precomputed using

Monte Carlo simulations for all the LDPC codes in the code set. When an image line is encountered with source distribution p_x 6= 0.9, the threshold values of the LDPC codes in the code set can numerically be computed for the given source distribution p_x. This is illustrated in Example 1.

Example 1: Assume that a threshold value $q_o^{\#}$ is found through Monte Carlo simulation at the source distribution, p_{x0} = P(X = 0). The conditional entropy H(X/Y) can be expanded as

$$H(X/Y) = H(p_{x0}) + H(q_0^{\#}) - H(p_{y0}) \quad (4)$$

where p_{y0} = P(Y = 0) = p_{x0} (1- $q_o^{\#}$) + (1 - p_{x0}) $q_o^{\#}$. Let $q_1^{\#}$ be the threshold value at source distribution p_{x1}. The following equation can be solved for $q_1^{\#}$ using the Newton-Raphson method.

$$H(p_{x1}) + H(q_1^{\#}) - H(p_{y1}) = H(X/Y) \quad (5)$$

where $p_{y1} = p_{x1}$ $(1 - q_1^{\#})$ + (1 - p_{x1}) $q_1^{\#}$, and $H(X/Y)$ is known from (4).

The equality sign is used in (5) assuming that $H(X/Y)$ is constant, with the number of simulated bits tending to infinity.

III. Decoding

As seen in Sec II-D, that the decision to encode an imageline using an LDPC syndrome code is based on the dataof the previous line i.e. $(i\text{-}1)^{th}$ line. Hence, an obviousway to decode LDPC coded lines is to assume the $(i\text{-}1)^{th}$ line as side information and decode the i^{th} line using SWdecoding. Such simple decoding may not be adequate to achieve good compression. Further, the spatial redundancy in the image is not exploited in an effective way. Hence, we propose context modelling at the decoder to make use of the spatial redundancy. In the following sections, we describe the details of context modelling.

The facsimile image scan lines can be treated as nonuniform sources, and hence, decoding strategy presented in [9] can be followed. In [9] the authors modified the decoding algorithm proposed in [16] for non-uniform sources and incorporated the source distribution while initializing the edges originating from the variable nodes of *bipartite graph* [17]. They modelled the side information as the output of an equivalent BSC with crossover probability equal to q.

Let $x^i,\ y^i \in \{0,\ 1\}$ be the realization of the non-uniform source X to be compressed with Y as side information. Both x_i, y_i belong to the i^{th} variable node, and the edges emanating from the variable node are initialized as

$$L_{i,0} = \log \frac{p(x_i = 0/y_i)}{p(x_i = 1/y_i)} = \log \frac{p(y_i/x_i = 0)P(x_i = 0)}{p(y_i/x_i = 1)P(x_i = 1)}$$
$$= (1 - 2y_i) \log \frac{1-q}{q} + \log \frac{p_0}{1-p_0}, \qquad i = 1, 2, ..., n \quad (6)$$

where $q = Pr(X \neq Y)$, p_0 = source distribution, $L_{i,0}$ = initial LLR value of i^{th} variable node.

Here, by including the $log(\ p_0/<1-p_0>)$ term in (6), the source distribution is taken into account while initializing the initial LLRs.

As seen from (6), the information about the source distribution $log(\ p_0/<1-p_0>)$ is added to channel information ((1-2y_i) log <1−q>/q) while initializing the variable nodes.

A. Context Modelling at Decoder

In JBIG and JBIG2, context-based coding is performed in conjunction with arithmetic coding. In this paper, a similar type of context modelling is used to improve the performance of the LDPC decoder. The initialization of LLR values in LDPC decoding is based on context modelling. The template used for context modelling is shown in Fig. 2. We would like to emphasize that the template used in this paper is neither an optimized one nor have we made any effort to optimize the same. We selected this template to explain the overall system concept; however, further research is warranted for the optimization of the template. The pixels marked '×' are used for the probability estimation of the pixel to be coded; the latter is marked '?'. In arithmetic coding, all the previous lines as well as the pixels to the on left of the pixel to be coded are available for context. In case of LDPC decoding, the whole block is decoded iteratively, and the final output is given by the wholly decoded block. In our case, each decoded image line represents the output from the LDPC decoder after the image line is successfully decoded or after the maximum number of iterations is reached. Hence, we cannot use the pixels to the left for context modelling. To overcome this difficulty, a novel context-based LDPC decoding is introduced in the Sec III-B.

In this paper, the adaptive context probabilities are derived from the previously decoded image lines above the present line. The strategy used for adaptive context modelling is similar to that used in case of JBIG [13] and is outlined below.

Let Z be an unknown pixel and C be the context. The probability of Z can be estimated as follows [12],

$$P(Z = 1/C) = \frac{n_1 + \triangle}{n + 2\triangle}$$

where n_1 is the number of times the pixel Z is 1 in the context of C, n is the number of times the context occurs, Figure 2. Template for Context Modelling in LDPC Decoding and 4 is a small constant taken as 0.5. The small value of 4 allows the process to start (i.e. $n_1 = n = 0$), and as n becomes large, the bias caused by 4 becomes negligible. Interested readers may find more information and improvements about the adaptive context modelling in [11, Chapter 11]. The estimated probabilities are utilized for initializing the edges originating from the variable nodes of the bipartite graph. Now, the $Li,0$ in (6) can be directly derived from the context probabilities as

$$L_{i,0} = \log \frac{p(x_i = 0/C)}{p(x_i = 1/C)} = \log \frac{1 - P(Z = 1/C)}{P(Z = 1/C)} \quad (7)$$

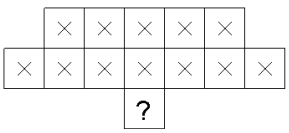


Fig. 2. Template for Context Modelling in LDPC Decoding

One can note that in (7), the initial LLR values are directly obtained from the context probabilities, whereas in (6), the initial LLR is derived from the crossover probability and source distribution. This implies that information about the crossover probability (q) and source distribution (p_0) is not required in context-based decoding.

B. Context-based LDPC Decoding

As already mentioned in Sec III-A, the pixels to the left side of Z cannot be used for context modelling as the whole image line is decoded simultaneously in LDPC syndrome

decoding. We have introduced a novel context-based LDPC iterative decoding to overcome this problem. The template for context-based LDPC decoding is shown in Fig. 3. Here, pixels to both the left and right of Z (= ?) are available for context modelling, as the entire image line is decoded simultaneously. Now the key issue that remains is the utilization of the context. In the initial iterations, soft information about the pixel values is not reliable and cannot be used. After the final iteration, the output is eventually obtained. Hence, we have incorporated weighted context information in LDPC iterative decoding. As the number of iterations increases, more weight is given to the context information derived from the pixels to both the left and right. To this end, a separate record of context probabilities is maintained with respect to the context template shown in Fig. 3. The context probabilities are derived from the previously decoded image lines as described in Sec III-A.

In context-based LDPC decoding, after every iteration, a hard decision has to be taken on the image line (LDPC codeword) to form the context. The LLR values are estimated from this context, and the initial LLR values are updated for every iteration as follows.

$$L_{i,0} = L_{i,0} + \frac{t}{T}\log\frac{1 - P(Z = 1/C)}{P(Z = 1/C)},\ t = 0, 1, 2, ..., T$$

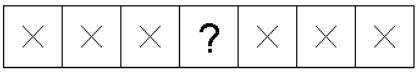


Fig. 3. Template for context used in LDPC decoding

where t is the iteration number and T is the maximum number of iterations allowed. Here, Z and C refer to the pixel to be coded (marked '?') and the context with respect to the template in Fig. 3, respectively.

IV. Experimental Setup and Simulation Results

In this section, we provide additional details, practical aspects, and realization of the building blocks described in Sec.II. The CCITT test images are used for testing the algorithm and for comparison purposes. The test images are of size 2376 × 1728 and are available online [18]. The thumbnail images of CCITT test images are shown in Fig. 4.

A. Variable Rate LDPC Codes

Different LDPC matrices (codes) are designed using variable number of rows and fixed number of columns; the number of columns is equal to 1728, which is equal to the image width. The (3, p) regular variable rate LDPC matrices are constructed using the PEG algorithm [19]. For simulation purposes, we have designed 61 LDPC codes for the LDPC code set. This accounts to 6 bits per image line in the header, which transmits the mode selection information. The '0' mode for $q = 0$ (i.e. when the present image line is the same as the previous one); 1 – 61, for different LDPC codes; 62, for VLC coded image line; and 63, for raw data. The LDPC codes are constructed using 96 – 1056 number of rows; the number of rows increases in steps of 16. Therefore, the bit rates range from 0.056 – 0.61 bits/pixel. The computed simulation threshold ($q^{\#}$) values range from 0.000579 for the (1728, 1632) LDPC code to 0.294 for the (1728, 672) LDPC code. It is observed that the inbetween values increase nearly in a parabolic fashion. For simplicity, we have selected regular LDPC codes with fixed step size. However, there is good scope to optimize LDPC codes in various other ways. Optimization may be addressed at several fronts; however, it is left for future research.

Numerous variable rate LDPC codes have been presented in literature, for example in [20], [21], [22], [23], [24]. Particularly, the variable rate LDPC code presented in [21] is very attractive in the present context because it presents a system architecture for LDPC codes that allows the dynamic switching of LDPC codes within the encoder and decoder without any hardware modifications. Further, it supports the codes that span a wide range of lengths and code rates, without compromising on the coding efficiency.

B. Variable Length Codes

For variable length coding, the well-proven modified Huffman (MH) codes [11] used in CCITT group 3 facsimile standard were followed. The MH code tables may be found in [25]. Better and more efficient VLCs will be explored in future research

C. Mode Selection for Context-based Decoding

The mode selection algorithm described in Sec II-D is suitable when the previous line is considered as side information and the present line is decoded by LDPC syndrome decoding. Context modelling is not considered in such mode selection. Further, the LDPC codes are analyzed according to the principles of Slepian-Wolf coding, i.e. their threshold values are determined using synthetically generated side information at a given source distribution. Hence, selecting an LDPC code with context modelling is difficult.

Intuitively, we assume that context-based decoding should improve the compression performance. To improve the compression performance, one should select an LDPC code with a high threshold value, while keeping the code rate high (resulting in a low bit rate). This is possible when the nonuniformity of source distribution is high. This indicates that context modelling effectively increases the source distribution. With this intuitive observation, we define an *overloading factor* to account for the context modelling at the decoder. The overloading factor *Fij* is used in the following equation to update the virtual source distribution *pCx* of source *X*, that matches the context modelling.

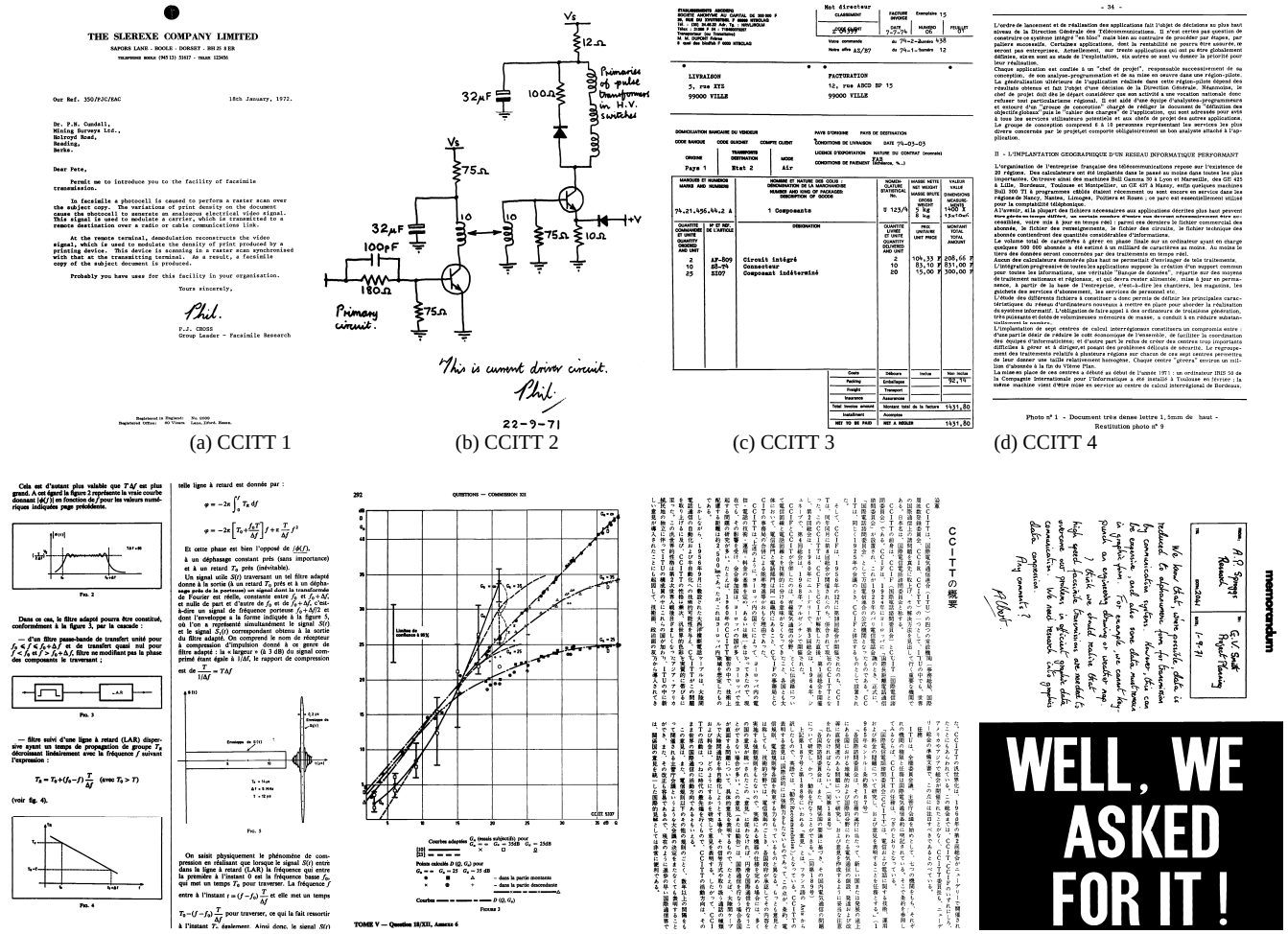
$$p^{C}_{x} = p_x + F_{ij}(1 - p_x) \qquad 0 \le F_{ij} \le 1$$

where *i, j 2 f*0*,* 1*g* and are defined as follows

$$i = \begin{cases} 0 & No\ Context\ Modelling\ at\ Decoder \\ 1 & With\ Context\ Modelling\ at\ Deocder \end{cases}$$

$$j = \begin{cases} 0 & No\ Context\ Modelling\ in\ LDPC\ Decoding \\ 1 & With\ Context\ Modelling\ in\ LDPC\ Decoding \end{cases}$$

The mode selection module selects an LDPC code assuming the virtual source distribution to be p^{C}_{x}. Because p^{C}_{x} is higher than p_x, a high rate LDPC code with high threshold is selected by the mode selection algorithm. The value of F_{ij} depends on the performance gain expected from context modelling.



(a) CCITT 1 (b) CCITT 2 (c) CCITT 3 (d) CCITT 4

(e) CCITT 5 (f) CCITT 6 (g) CCITT 7 (h) CCITT 8

Fig. 4. CCITT test images

D. Simulation Results

The proposed algorithm is applied to the CCITT images with each image is encoded for different overloading factors and decoded. The overloading factors listed in Table I represent the highest overloading factor for which near lossless decoding is achieved. The first column lists the compression ratios obtained when the overloading factor is zero and no context model was used at the decoder. Obviously, no overloading can be used when context modelling at the decoder is disabled. On an average, an increase of 0.60 in the compression ratio is obtained with context modeling(combined context-based LLR initialization and context-based LDPC decoding) as compared to that obtained with no context. Context-based LDPC iterative decoding itself increases the compression ratio by 0.42 in comparison with no context.

The proposed algorithm outperforms CCITT group 3-1D (fax 3-1D) coding by a convincing margin. With respect to fax 3-2D, this algorithm is sometimes superior and some times inferior, resulting in a mixed performance. For images, a BER of 10−5 or less can be considered as a near lossless reconstruction. The BER is computed as follows.

$$BER = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\mathbf{A}(i,j) \oplus \hat{\mathbf{A}}(i,j)$$

where b**A** is the reconstructed image. In some cases, the BER does not fall below 10−5 even when the overloading factor is zero. This condition is indicated with a footnote in Table I and considered as lossy compression. However, the reconstructed image appears to be visually lossless. It is observed that this situation occurs whenever there is dense text in the image. Such situations will be analyzed in our future research.

TABLE I: COMPRESSION RATIOS FOR CCITT TEST IMAGES AND CORRESPONDING OVERLOADING FACTORS

Image	CR[1]	F_{01}	CR	F_{10}	CR	F_{11}	CR
CCITT 1	16.02	*0.65*	16.82	*0.59*	16.63	*0.73*	17.16
CCITT 2	18.10	*0.86*	18.43	*0.82*	18.37	*0.87*	18.54
CCITT 3	08.63	*0.62*	08.94	*0.57*	08.92	*0.63*	08.96
CCITT 4	4.78[2]	0.32	4.89	*0.35*	04.92	*0.60*	05.18
CCITT 5	7.90[2]	*0.66*	08.26	*0.62*	08.22	*0.75*	08.50
CCITT 6	11.01	*0.75*	12.05	*0.65*	11.78	*0.75*	12.16
CCITT 7	4.51[2]	*0.34*	04.59	*0.27*	04.57	*0.50*	04.72
CCITT 8	10.35	*0.81*	10.65	*0.72*	10.54	*0.85*	10.92

1CR with $F00 = 0.00$ 2BER > 10−5

TABLE II: COMPARISON RESULTS OF THE PROPOSED ALGORITHM

Image	Fax 3-1D	Fax 3-2D	Fax 4	Proposed Algorithm
CCITT 1	13.70	14.21	17.03	17.16
CCITT 2	14.90	16.59	22.36	18.54
CCITT 3	07.89	09.27	10.99	08.96
CCITT 4	04.75	05.31	05.56	05.18
CCITT 5	07.51	08.77	10.16	08.50
CCITT 6	10.00	12.16	15.69	12.16
CCITT 7	04.82	05.35	05.63	04.72
CCITT 8	08.17	10.40	13.60	10.92

V. CONCLUSION

We investigated the compression of facsimile images under the DSC paradigm. It is a single source near-lossless image compression with side information (for Slepian-Wolf decoding) derived from the previously decoded image lines.

The side information was generated using context-based modeling at the decoder. Further, we have introduced a novel weighted context-based LDPC iterative decoding in addition to realizing context-based initialization of log-likelihood ratio (LLR) values in LDPC decoding. The combination of these techniques improved the compression performance considerably. On an average, an increase of 0.60 in the compression ratio is obtained with context modelling. The performance of the overall system is extremely compatible with fax3 (CCITT group 3) compression. As the proposed method is based on DSC principles, it is inherently error resilient and affords a system with a low-complexity encoder.

REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources", in IEEE Trans on Information Theory, Vol. 19, No. 4, Jul 1973, pp: 471-480.

[2] Z. Xiong, A. D. Liveris and S. Cheng "Distributed Source Coding for Sensor Networks" in IEEE Signal Processing Magazine, Sept 2004, pp: 80-94

[3] C. Gikici, et al, "Distributed Source Coding of Still Images", in 13th European Signal Processing Conference (EUSIPCO 2005), Antalya, Turkey, Sept 4-8, 2005.

[4] J. E. Vila-Forcen, et al, "Distributed Single Source Coding with Side Information", in Proc. of IS&T/SPIE 16th Annual Symposium: Electronic Imaging 2004, Image Processing.

[5] D. Varodayan, A. Aaron and B. Girod, "Exploiting Spatial Correlation in Pixel-Domain Distributed Image Compression", in Proc. International Picture Coding Symposium, Beijing, P.R. China, Apr 2006.

[6] M. I. Kroll, "Wireless Fax Machine", available online on http://www.invention.net/ugi.htm

[7] J. Kroll and N. Phamdo, "Source-channel Optimized Trellis Codes for Bi-tonal Image Transmission over AWGN Channels" in IEEE Trans on Image Processing, Jul 1999, pp: 899-912.

[8] R. G. Gallager, "Low-Density Parity Check Codes", IRE Tran. Inform. Theory, Vol. IT-8, Jan 1962, pp:21-28.

[9] R. K. Bhattar, K. R. Ramakrishnan and K. S. Dasgupta, "Simulation Study of LDPC Codes for Nonuniform Sources with Side Information in Slepian-Wolf Coding" in NCC 2010, Vol., No., IIT Chennai, India, 29-31 Jan 2010, pp:1-5.

[10] R. K. Bhattar, K. R. Ramakrishnan and K. S. Dasgupta, "Analysis of LDPC Codes for Compression of Nonuniform Sources with Side Information Using Density Evolution" in DCC 2010, Vol., No., 24-26, March 2010, pp:522-522.

[11] N. Memon and K. Sayood, "Facsimile" in Handbook of Communication, Edited by J. D. Gibson, CRC Press 1996.

[12] "Document and Image Compression", Edited by M. Barni, CRC Press 2006.

[13] "JBIG Homepage", http://www.jpeg.org/jbig/index.html

[14] "JBIG2", http://jbig2.com/

[15] Y. Sun and J. Li, "Distributed Source Coding With Context Modeling" in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), 14-19 May 2006

[16] D. Liveris, Z. Xiog and C. N. Georghiades, "Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes", in IEEE Communications Letters, Vol. 6, No. 10, Oct 2002, pp: 440-442.

[17] W. E. Ryan, "An Introduction to LDPC Codes", in "Coding and Signal Processing for Magnetic Recording Systems" Edited by E. M. Kurtas and B. Vasic, CRC Press 2005.

[18] CCITT test images available online http://tc18.liris.cnrs.fr/code_data_ set/2D_binary/Scanned_Text/

[19] David MacKay's Gallager code resources, "Source code for Progressive Edge Growth Parity-check Matrix Construction", available online. http://www.cs.toronto.edu/~mackay/S0.html#PEG _ECC.html.

[20] D. Varodayan, A. Aaron and B. Girod, "Rate-Adaptive Distributed Source Coding using Low-Density Parity-Check Codes", available online http://ivms.stanford.edu/~dsc/publications/varodayan_ 05asilomar_rateadaptiveLDPC.pdf

[21] P. Fewer, M. F. Flanagan and A. D. Fagan, "A Versatile Variable Rate LDPC Codec Architecture" in IEEE Transactions On Circuits and Systems Part 1 Regular Papers, Vol 54, No. 10, 2007, pp: 2240-2251.

[22] M. Baldi, G. Cancellieri and Franco Chiaraluce, "Variable Rate LDPC Codes for Wireless Applications", in Proc. of the 2006 International Conference on Software, Telecommunications and Computer Networks (SOFTCOM 2006), Paper S9-6017-2909, Split, Dubrovnik, Croatia, 29 Sept 1 Oct, 2006.

[23] L.o Dinoi, F. Sottile and S. Benedetto, "Design of Variable-rate Irregular LDPC Codes with Low Error Floor" in IEEE International Conference on Communications, 2005, (ICC 2005), Vol. 1 pp: 647-651.

[24] N. Pandya, B. Honary, "Variable-rate Capacity-approaching LDPC codes for HF Communications" in 10th IET International Conference on Ionospheric Radio Systems and Techniques, (irst 2006), pp: 123-127.

[25] L. C. Mai, "Introduction to Image Processing and Computer Vision", available online www.netnam.vn/unescocourse/computervision/ computer. htm

Raghunadh K Bhattar was born on 19th June 1962 in Parvatipuram (AP), India. He received the B.Sc(Engg) from REC, Kurukshetra, India and M.Sc(Engg) from Indian Institute Science (IISc), Bangalore, India in 1985 and 2000 respectively. Presently he is pursuing Ph.D degree in Electrical Engineering IISc. He has joined Space Applications Centre, ISRO in 1990. He is actively involved in the design and development of Turbo codec on FPGA for ground based SATCOM systems and contributed to MPEG2 video codec, JPEG and JPEG2000 systems. Currently, he is a Project Manager, IRNSS Signal Simulator. He has reviewed several papers for IEEE Trans. on Geoscience and Remote Sensing, IEEE Geoscience and Remote Sensing Letters and Conference papers. His current research interest are digital signal processing, image and video coding, multimedia, wavelets, channel coding, wireless technologies and navigation technologies. He is a member IETE India.

K.R.Ramakrishnan (M'00) received the B.E., M.E., and Ph.D degrees in Electrical Engineering from the Indian Institute of Science, Bangalore, India in 1974, 1976 and 1983 respectively. He is currently a Professor with Department of Electrical Engineering, Indian Institute of Science. His research interests include image processing, computer vision, medical imaging, water marking and multimedia communication.

K.S.Dasgupta received the B.E and Post graduate course in computer science with Honors from Jadavpur University, India in 1972 and 1973 respectively. He received the Ph.D degree from Indian Institute of Technology, Mumbai, India in 1990. He has joined Space Applications Centre, ISRO, India in 1974. Since then he has contributed significantly in the field of image processing and satellite communications. As a Group Director, Advanced Digital Communication Technology Group (ADCTG), he was instrumental towards design and development of PC based multimedia system for satellite based distance education. He held the posts of Deputy Director and Director in ISRO and contributed immensely. Presently he is a Director IIST, Thiruvananthapuram, India. His current areas of research interest addresses digital signal processing, digital image processing, computer architecture and digital communications. He is a senior member IEEE, Fellow IETE India.