Source and Channel Coding Techniques for faithful transmission of Digital Mammograms

by Danoush Hosseinzadeh, April Khademi and Sri Krishnan
Ryerson University, Toronto, ON

Abstract

This paper examines source and channel coding techniques that are necessary for reliable transfer of digital mammograms over a wireless AWGN channel. First, in order to maximize bandwidth efficiency while maintaining the input signal's entirety, lossless source compression techniques are investigated. Particular attention is paid to compressors that produce the highest degree of compression. Second, to ensure reliable data transfer over a noisy wireless channel, error correction and detection mechanisms are discussed focusing on schemes that achieve the lowest bit error rates. Specifically, for experimental purposes, digitized mammograms saved as Portable Grey Maps (PGM) were used. Prior to choosing the optimal coding arrangement, the implications of the source and channel coding algorithms on PDAs and computers will be examined. Major concerns are computational complexity and encoding and decoding delays.

Sommaire

Cet article examine les techniques de codage source et de canal qui sont nécessaires pour le transfert fiable des mammographies numériques sur un canal sans fil à bruit additif blanc gaussien (AWGN). D'abord, afin de maximiser l'utilisation de la bande passante tout en maintenant l'intégralité du signal d'entrée, des techniques de compression source sans perte sont étudiées. Une attention particulière est prêtée aux compresseurs assurant le taux de compression le plus élevé. En second lieu, pour assurer un transfert fiable de données à travers un canal sans fil affecté de bruit, la correction d'erreurs et les mécanismes de détection sont discutés en focalisant sur des configurations de codage réalisant les plus bas taux d'erreurs sur les bits. Spécifiquement, pour des fins expérimentales, des mammographies numérisées et sauvées sous format PGM ont été utilisées. Avant de choisir la configuration optimale de codage, les implications des algorithmes de codage source et de canal sur les assistants numériques personnels (PDAs) et des ordinateurs seront étudiées. L'objet principal de l'étude sera la complexité du calcul et les retards dans le codage et le décodage.

1.0 Introduction

Telemedicine is a fairly new and emerging field of engineering that is dedicated to providing hi-tech technologies to the medical profession. This paper seeks the most favorable method in which digital mammograms can be prepared for transfer over a wireless link. This transfer could be over a radio channel and may be between two computers or from a computer to a piece of mobile equipment.

The intended beneficiaries of this research are professionals who may need to transfer sensitive, medical information; in particular medical images (i.e. digitized x-rays) and physiological signals for an expert referral, and in emergency ambulatory situations.

The mobile unit of choice that is under investigation is the Personal Digital Assistant (PDA), which is easy to use, portable and is a powerful computing device with a high definition display. As a result, this research will consider the operability of any channel or source coding technique on these types of devices.

The reliability of data transmission will be simulated with real-world medical mammograms (x-ray images of the breast). The images investigated are from the MIAS database, which contains numerous mammograms that are normal, benign or malignant (Figure 1).

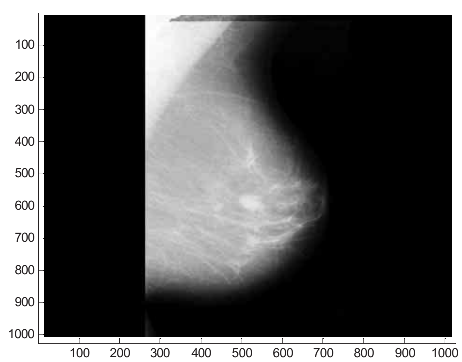

Figure 1: PGM sample of digitized mammogram (benign case)

2.0 Source Compression

Since the mammograms in question are to be later interpreted by a diagnosing physician, it is important to retain the image's accuracy and therefore lossless compression schemes will be investigated. An optimal lossless compression scheme will reduce the input image size significantly. By doing so, it will compensate for the additional redundant bits inserted by the channel encoder. Additional considerations such as computational complexity, encoding delays, transmission overhead and whether these compression techniques are practical for various computing devices will be investigated prior to deeming a source encoder favorable.

2.1 Source Description

When deciding which compression scheme is optimal for a desired application, it is first necessary to characterize and understand the unique features of the source. Identifying the source characteristics will allow an appropriate source encoder to exploit those qualities. Sources may be grouped into two categories:

1) Memoryless Source - a source that emits symbols that are statistically independent of one another.
2) Memory Source - a source that emits symbols that are dependant on any number of its previously emitted symbols.

When considering a digital image, it is highly unlikely that adjacent pixels are going to be perfectly independent of one another. This is because regions in a digital image are usually concentrated with one color and gradually blend into another. Therefore, it is concluded that each pixel exhibits a high degree of correlation with its neighbor pixel, and as a result, the digital image inputs can be modeled as a source with memory.

After some investigation, it was found that digital images prove to be exceptionally correlated with their immediate neighbors [1]. This leads to the deduction that a first order Markov model would be an accurate description of the source.

2.2 Adaptive Huffman

The adaptive Huffman compressor is a modification of the conventional Huffman algorithm. Instead of gathering symbol statistics first, the encoder and decoder use a dynamic tree that counts the probability of occurrence and assigns a codeword to each symbol, in real time. This is one of the major advantages of any adaptive Huffman scheme. It does not need to first scan the entire file to gain knowledge of the input file's characteristics, which significantly reduces the total encoding time for large files. Therefore, the only delay arises from the computational complexity of the algorithm. For the Faller, Gallager, and Knuth (FGK) algorithm, D.E Knuth has proved the complexity to be O(l) where l is the current length of the codeword [2].

Additionally, both the encoder and decoder manage their own information about the input message and do not require a look up table, unlike static Huffman codes. As a result no additional information needs to be transmitted with the encoded symbols allowing for maximum bandwidth utilization.

A downfall of any Huffman compressor is due to the fact that maximum compression is achieved when the symbol probabilities are a power of 1/2 [3].

2.3 Arithmetic Coding

Arithmetic compressors are composed of two parts: the arithmetic coder and the Markovian data model. The arithmetic coder assigns a floating point interval between 0 and 1 for each symbol. The interval endpoints are allocated by dividing the original encoding interval according to a specific algorithm that depends on the source probability distribution, which is described by a Markovian model.

Arithmetic coders are ideal for large symbol sets as the floating point interval can be divided as many times as the computing processor allows. Another advantage of arithmetic compression arises from the fact that the Markovian model takes into consideration the correlation between symbols, thus allowing for higher compression.

A major disadvantage of any arithmetic coder is its computational complexity. Some scholars say that for arithmetic coders that incorporate a Markovian data model, the complexity is at least O(n3), where n is the length of the codeword [4]. Furthermore, because a probability model is required, the input file must be scanned twice to be encoded and the symbol mapping scheme must also be transmitted along with the codewords. As a result, delay and transmission overhead are introduced.

2.4 Lempel-Ziv Welch (LZW) Coding

LZW compression is achieved using a dictionary based approach. As input symbols arrive, the encoder checks the dictionary. If the symbol can be located within the dictionary, the related codeword is assigned and transmitted. Otherwise, the new symbol is added to the dictionary and is assigned a unique codeword. Since codewords are appointed to a given symbol only if it is found in the dictionary, LZW codes rely heavily on repeating patterns. This can be good or bad depending on the application. If the input data is highly correlated, LZW compressors will find a large number of repeating patterns which will produce a high degree of compression. However, if there are only a few recurring patterns, the dictionary will be indefinitely long and the achieved compression will not be significant.

Similar to adaptive Huffman schemes, prior to encoding, LZW codes do not require any knowledge of the symbol's statistical characteristics and both the encoder and decoder maintain their own dictionaries. As a result, encoding only requires one scan of the file and delays are caused primarily as a consequence of computational complexity. If a message of length u is compressed to a length of n, when considering a pattern of length m, to find all R occurrences of this pattern in the message takes O(2m+mn+Rmlog(m)) time for the worst case scenario. On average, the worst case drops to O(2m+(n+R)log(m)), which will still reach especially high values for long message and pattern lengths [5].

2.5 Experimental Results

To test the performance of various lossless source encoders, several digital mammograms supplied by the MIAS database were used as input messages. In particular, the source coding algorithms that were experimented with were adaptive Huffman codes, LZW codes with 12 bit and 15 bit dictionaries and arithmetic coders with order-0 and order-1 probability models. A coder with an order-n probability model assumes that the present symbol is statistically dependant on the previous n symbols emitted from the source. As previously discussed, the images in question were found to exhibit exceptional correlation with their immediate neighbour, so the data model need not be higher than an order-1. A desktop computer with a 1.4 GHz processor was the computing device where all tests were conducted.

Average compression ratios for 30 mammograms (in bits per pixel) can be found in Figure 2.

The average first order Markov entropy over all images was found to be 1.5381 bits/pixel and was calculated by

$$H(x) = \sum_{x \in X} P(x) \sum_{y \in Y} P(y \mid x) \log_2 \left[\frac{1}{P(y \mid x)} \right] \qquad (1)$$

Where:

x and y are the present and past integer pixel values respectively.

Using the found entropy results, the efficiency for the previously listed source coders was easily found and are illustrated in Figure 3.

Upon examination of Figures 2 and 3, arithmetic coding with a first order probability model gave rise to the lowest compression ratio and

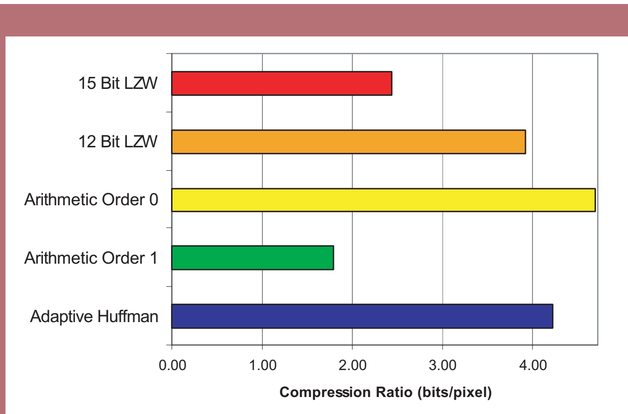

Figure 2: Average compression ratios in bits/pixel

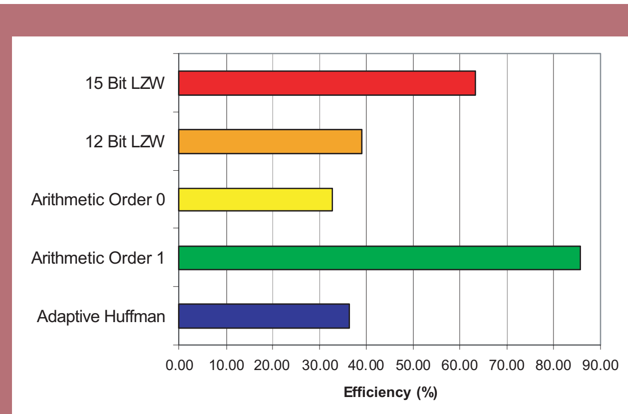

Figure 3: Efficiency results

the highest efficiency. LZW with a 15 bit dictionary also produced desirable compression and efficiency results, even though they did not supersede the results produced by the arithmetic order-1 compressor. However, as previously stated, there are many other concerns that must be considered when choosing the optimal source compression scheme for any application. Such considerations include computational complexity, encoding and decoding speed and whether it is possible to execute these compressors on different devices.

The adaptive Huffman compressor is not a very complex algorithm O(n)) and does not require a probability model. For these reasons, the encoding and decoding delay was minimal and was noted to be in the order of milliseconds. Although this was confirmed using a 1.4 GHz processor, it is very easy to deduce that because of the small number of computations required and the need for only a single file scan, similar results could be reproduced on a PDA that uses a 400 MHz processor. Additionally, adaptive Huffman codes do not transmit the symbol mapping scheme along with the encoded symbols, which permits for maximum bandwidth utilization.

Although adaptive Huffman codes performed exceptionally well in all other aspects, the compression it achieved was very poor. Its heavy reliance on symbols that occur statistically in powers of 1/2 is the reason for the inadequate compression results.

Arithmetic coders are a versatile compressor as they can account for a Markovian data model and therefore data correlation. The advantages are visible from the results shown in Figures 2 and 3. Since the digital images have been found to exhibit correlation in a first order Markov fashion, the immense benefits of considering a Markovian data model can be seen when comparing the exceptional compression characteristics of an arithmetic order-1 coder with the poor results of an arithmetic coder using a order-0 probability model.

Arithmetic coding is also a flexible compression scheme because it can handle large symbol sets. This is true because the symbol mapping scheme is dependent on a floating point interval and how many times it can be divided, which relies on the precision of the computing device. For computers and PDAs, which have processors that can maintain the precision of a number within 64 and 32 bits, respectively, large symbol sets are easily supported. The precision that these devices sustain also allows interval end points to be rather insensitive to truncation and rounding errors.

The major downfall of arithmetic coding is its complexity. As stated before, when an arithmetic coder considers a Markovian model for the source, which requires two scans of the input file, complexity is at least O(n3). This was confirmed experimentally by observing the encoding and decoding delays. Because arithmetic order-0 did not use a memory source model, it encoded and decoded rather quickly, whereas even on a 1.4 GHz processor, arithmetic order-1 took a few seconds. Since the arithmetic order-1 compressor gave rise to the most desirable results, its complexity is a major concern. For PDAs that operate at 400 MHz, there may be a longer delay than that which was found for the desktop computer. For a real time application, this delay may not be desired.

Because the LZW algorithm is based on finding repeating patterns, it can be very effective when compressing files that are highly correlated. In order to achieve high compression ratios, two things need to be satisfied: the input file must be large enough to be able to find a significant amount of pattern repetitions and the maximum buffer length in the dictionary must be longer than the period of the longer repeating patterns. For both LZW 12 and 15 bit, the message size was large enough, but for LZW 12 bit, the maximum buffer length was not long enough to realize the longer patterns. This was concluded upon observation of the fact that LZW 12 bit performed poorly with respect to LZW 15 bit. In fact, LZW 15 bit offered almost as good compression results as those of arithmetic order-1.

Although LZW 15 bit provided good results, it suffered from a noticeable encoding and decoding delay. As mentioned before, delays in the compression and expansion process is due to the average time O(2m+(n+R)log(m))) taken to find R patterns of length m, to result in a compressed file of size n. For large message sizes, which is the case here, and for patterns that are a maximum length of 2^{15} (for LZW 15 bit), this computational complexity becomes extremely high. When LZW was tested on a 1.4 GHz processor, the entire encoding and decoding process took a couple of seconds longer than that of arithmetic order1. Although LZW codes are computed in real-time and only require one scan of the file, the decoding delay may be more noticeable slower processors, like those of a PDA. When considering source compression schemes, one needs to decide whether the delay is worth the amount of compression that is achieved. Such a delay may be undesirable.

3.0 Channel Codes

There are many types of channel encoding schemes that can be used in digital communication systems such as convolutional codes, linear block codes (LBC), and various types of hamming codes. The performance of two of these techniques, convolutional codes and LBC, were investigated and tested in the transmission of digital mammograms in all white Gaussian noise (AWGN) channel.

3.1 Choosing a Channel Coder

Convolutional and linear block codes (LBC) were chosen because they can combat many types of channel impairments including AWGN. However, it will be demonstrated that convolutional codes have several distinct advantages over LBC, and because of this, convolutional codes are the focus of this section.

Convolutional encoders are simple to design and operate at very high speeds because they are composed of simple logic circuits. Also, convolutional codes can be designed with much more ease than LBC codes because no systematic procedure exists to aid in the design of LBC codes [6]. Additionally, efficient low rate convolution codes are widely used in many applications such as the Global System for Mobile Communication (GSM), IEEE 802.11x standards for wireless local area networks (WLAN) and NASA's deep space communication. This demonstrates that convolutional codes can operate on a variety of platforms including PDAs.

3.2 Convolutional Encoding

In general, any convolutional encoder can be described by the three parameters: n, k and K, where n is the number of output bits, k is the number of input bits per output set and K is the constraint length of the encoder which is equal to the number of memory location in the register. This is shown in Figure 4.

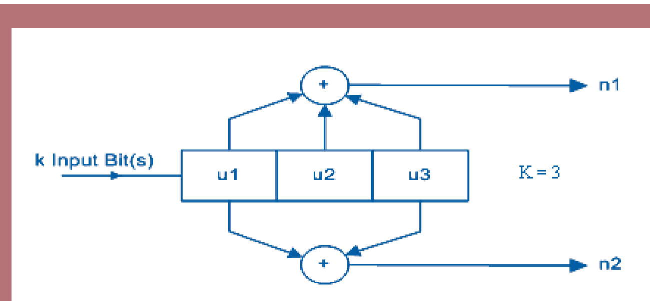

Figure 4: Convolutional encoder with encoding polynomials 7,5 (in octal), rate k/n and constraint length (K =3)

The rate or efficiency of the encoder is the ratio of the input bits to the output bits per sample time shown below:

$$Rate = Efficiency \quad (\eta) = \frac{input \;\; bits}{output \;\; bits} = \frac{k}{n} \qquad (2)$$

The encoding rate affects the bandwidth of the transmitted message significantly. For example, if k=1, then the rate (or efficiency) is equal to 1/n, which means that the encoder output is n-times the input. This is the primary reason that source compressors are often used in conjunction with convolutional coders as they compensate for the increase in file size. To reduce bandwidth requirements only encoding rates of 1/2 and 1/3 were considered. Furthermore, to significantly reduce the decoding complexity (memory as well as computations) only rate 1/n encoders were considered.

3.3 Viterbi Decoding

Although encoding is very fast and simple for any given set of specifi cations, convolutional decoders are considerably more complex. low constraint length (K) encoders, Viterbi decoding can drastically reduce complexity and can operate as fast as 100 Mbits/s [6]. In fact, because of its simplicity, Viterbi decoding is one the main advantages of convolutional encoding. Simplifications offered by Viterbi decoding include the use of hamming or Euclidean distances to measure error the received codeword, as well as reducing the memory requirements half via the introduction of survivor paths as is demonstrated in [8]. The main parameter that requires attention, in Viterbi decoding is decoding depth (D). The decoding depth determines the amount of error that can be corrected by the decoder. It also establishes the mem ory requirements of the decoder and can be expressed by

$$memory = D \times 2^K \tag{3}$$

Because Viterbi decoding relies on maximum likelihood probabilities, small values of D will produce poor Bit Error Rate (BER) results while large values of D will produce the best BER results. However, for large values of D, the decoder complexity increases significantly. Experimental results will determine the best decoding depth to be used for each encoder polynomial set. Furthermore, decoding complexity depends on the constraint length (K). We can see from equation 3 that decoder complexity and memory requirements grow proportional to 2K. Therefore, Viterbi decoding becomes impractical for large values of K (K>10) and in practice K is often kept small for this reason [6]. To minimize the decoder complexity, the encoder constraint length (K) was kept to below five.

3.4 Channel Simulation

The performance of convolution encoder polynomials are difficult to predict. To aid in the design and experimental process, a Viterbi decoder was designed with a built-in AWGN channel simulator so that BER results for any rate 1/n encoder could be calculated.

In order to calculate BER, prior to transmission into the AWGN channel, the simulator first performs polar-NRZ line coding techniques on the binary data. Polar-NRZ line coding techniques map binary 1 to +1 and binary 0 to -1, as can be seen in Figure 5. Upon reception, it is then the decoder's job to decode the noisy output from the channel. After doing so, the performance of the encoding polynomial can be measured based on decoded bit error rates. Assuming polar-NRZ line coding has been used prior to transmission; the theoretical BER of the received message can be calculated by

$$BER_{Polar-NRZ} = erfc\left(\frac{\sqrt{SNR}}{2}\right) \tag{4}$$

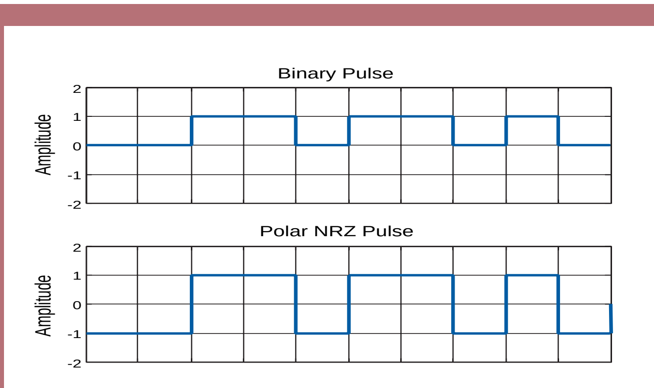

Figure 5: Binary waveform and its corresponding polar-NRZ waveform

3.5 Experimental Results

By limiting the encoder rate to 1/2 and 1/3 as well as limiting the con straint length (K) of the encoder to below five (as discussed earlier), there was a significant reduction in the memory requirements of the decoder, while simultaneously reducing the number of computations per decoded bit.

Many polynomial sets were tested and their performance was measured using the simulator. From these polynomial sets, the best rate 1/2 and polynomials were chosen for the final design. These polynomials and their characteristics are shown in Table 1, and their performance charac teristics are shown in Figure 6 for the case of 1/2 convolutional encoder after Viterbi decoding and for varying decoding depths (D). Figure accurately depicts two things: Signal Noise Ratio (SNR) versus BER improvement and also that longer decoding depths do not provide sig nificant BER improvement. Since Figure 6 does not provide sufficient comparative faetures, this figure is repeated as Figure 7 but without the uncoded channel.

Table 1: Convolutional encoder characteristics

Rate	Constraint Length	Polynomial 1	Polynomial 2	Polynomial 3
1/2	3	$x^2 + x + 1$	$x^2 + 1$	-
1/3	4	x^3	x + 1	1

We can see from Figures 7 and 8 that the chosen polynomials provide a significant BER improvement when compared with the uncoded message, which can also be seen on the same figures. Furthermore, while achieving this BER performance, the algorithms can still be implemented on both hand held and larger computers because of the significant reduction in complexity as discussed.

Lastly, it was found that longer decoding depths also improved BER performance of the decoder. However, from Figures 7 and 8 it can be seen that decoding depths of approximately 5K-7K produce comparable results to decoding depths of 50K for the rate 1/2 polynomial set. Similar results can also be seen for the rate 1/3 polynomial set. However, smaller decoding depths significantly decrease the memory requirements of the decoder and hence are preferred.

4.0 Conclusion

To prepare a digital mammogram for reliable transmission through a wireless channel, the image must first be compressed to minimize bandwidth usage and then a method of error detection and correction must be For the mammogram images tested, two compression schemes, arithmetic order-1 and LZW (with a 15 bit dictionary) produced the highest

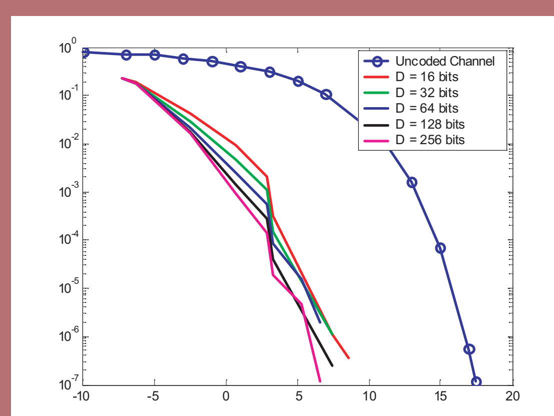

Figure 6: SNR vs. BER results for the rate 1/2 convolutional encoder after Viterbi decoding and for varying decoding depths (D)

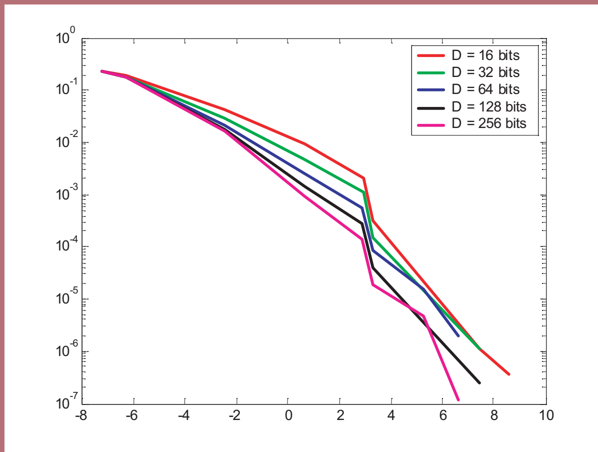

Figure7: SNR vs. BER results for the rate 1/2 convolutional encoder after Viterbi decoding and for varying decoding depths (D)

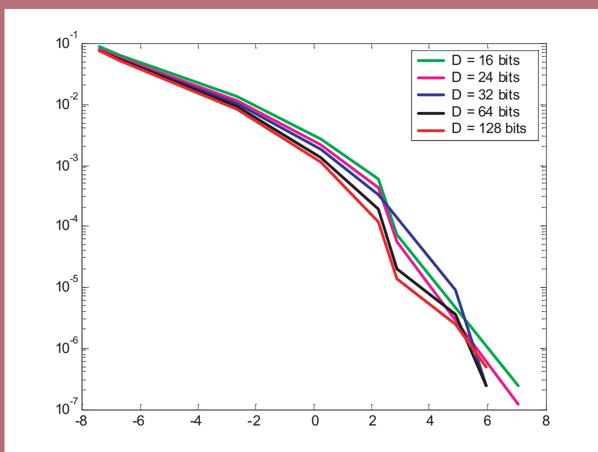

Figure 8: SNR vs. BER results for the rate 1/3 convolutional encoder after Viterbi decoder and for varying decoding depths (D)

compression results. Of the two, arithmetic order-1 achieved the highest compression ratio because it uses a Markovian data model which exploits the correlation between adjacent pixels. This exploitation increases the computational complexity which translates into longer encoding and decoding delay with respect to LZW (with a 15 bit dictionary) which operates in real-time. Arithmetic order-1 will also be less bandwidth efficient than LZW (with a 15 bit dictionary) because the receiver needs the encoder's data model, thus creating transmission overhead. Although an arithmetic order-1 compressor may be less bandwidth efficient and introduces longer encoding and decoding delays, it is still well warranted for image transmission as more compression will allow for better error correction and detection.

After image compression, a channel coding scheme must be used to ensure reliable data transfer. Convolutional codes were chosen as the channel coding technique because of their ability to correct various types of channel impairments and also because efficient low rate convolutional coders are widely being used today in many technologies. To further reduce computational complexity, only rate 1/2 and 1/3 encoders were considered with constraint lengths of less than five which enables easy Viterbi decoding. These simplifications allow for the decoder to operate on a variety of platforms.

Viterbi decoding was chosen because it is widely used for convolution codes and because of its simplicity. The optimal decoding depth (D) was found to be 5K-7K, where K is the encoder constraint length.

5.0 References

[1]. Liang, Tao., "Effect of Scan Order on the Efficiency of Predictive Coding." <http://ise.stanford.edu/class/ee392c/demos/liang> Stanford University (1998).

[2]. Hirschberg, Dan., "Adaptive Huffman Coding." <http://www.ics.uci.edu/~dan>, (2004)

[3]. Moffat, A.; Neal, R.; Witten, I.H., "Arithmetic Coding Revisited." IEEE Proceeding Data Compression Conference, 1995. March 1995. Pages 202 - 211.

[4]. Chuang, Isaac L., Modha, Dharmendra S., "Reversible Arithmetic Coding for Quantum Data Compression." IEEE Transactions on Information Theory. Volume 46, Issue 3. May 2000. Pages 1104 -1116.

[5]. Navarro, Gonzalo, "Regular Expression Searching Over Ziv-Lempel Compressed Text." University of Chile. 2001.

[6]. Lathi, B. P., "Modern Digital and Analog Communication Systems, 3rd Edition". Oxford University Press, New York, 1998. Pages 736 - 752.

[7]. Sklar, Bernard, "Digital Communications, Fundamentals and Applications, 2nd Edition". Prentice Hall PTR, New Jersey, 2001. Pages 401 - 405.

About the authors

Danoush Hosseinzadeh received a B. Eng in 2004 in Electrical Engineering from Ryerson University where he also plans to pursue a Master's degree. His research interests are mainly related to signal processing and cellular communications. Presently, he is a NSERC/USRA award winner and is researching a variety of time-frequency techniques suitable for signal analysis and feature extraction. Recently, Danoush has received the award of "Best Undergraduate Technical Paper" at the ICUE 2004 conference.

April Khademi is a recent graduate from an Electrical Engineering program at Ryerson University where she has received numerous awards for outstanding achievements. One of the most notable awards, received at ICUE 2004, was for "Best Undergraduate Technical Paper." In the Fall, she will be pursuing her Master's degree in the field of Biomedical Engineering. Her research interests are primarily in the area of signal processing and telecommunications. She is currently working at a company that develops computer peripherals that interface with gaming software.

Sridhar (Sri) Krishnan received a Bachelor of Engineering degree in Electronics and Communication in 1993 from Anna University, Chennai, India, and the M.Sc. and the Ph.D. degrees in Electrical and Computer Engineering from the University of Calgary, Calgary, Alberta in 1996 and 1999 respectively. He is at present an Associate Professor with the Department of Electrical and Computer Engineering at Ryerson University, Toronto, Ontario. His research interests are in the areas of adaptive signal processing, biomedical signal analysis, and multimedia communications. He is the Director of the Signal Analysis Research (SAR) group at Ryerson University that conducts research in medical image and signal analysis research (miSAR), and multimedia information and signal analysis research (MISAR).