

Estudo e Implementação Computacional de um Sistema de Comunicação Digital

Da Silva Junior, A. C.

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
amarocsjunior@gmail.com

Alcoforado, M. L. M. G.

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
mariadelourdesalcoforado@yahoo.com.br

Resumo

Este artigo visa apresentar o desenvolvimento de um software que permite dar subsídios ao ensino da Engenharia na área de Telecomunicações. O artigo expõe, além da busca por melhores resultados de demonstrações e melhor didática de apresentação do software, o estudo e implementação de um sistema de comunicação digital, abordando três subtemas principais: codificação-decodificação de fonte, codificação-decodificação de canal e canal corrompido com ruído aditivo Gaussiano Branco (AWGN). Ao analisar-se o que se tem hoje como apoio ao ensino de Engenharia, percebemos uma não evolução do velho sistema quadro-caderno. Este artigo apresenta uma nova opção de dinâmica de ensino.

Abstract

This paper aims to shows a software development that allows give subsidies to Engineering teaching in Communication area. The paper exposes, beyond the search for better results of statements and better didactic of software's presentation, the study and implementation of a digital communication system, addressing three main subthemes: source coding-decoding, channel coding-decoding and corrupted channel with Additive White Gaussian Noise (AWGN). When analyzing what we have today to support the teaching of Engineering, we realize a non-evolution of the old frame blackboard-notebook. This paper shows a new option for dynamic teaching.

1 Introdução

Em 1948, no Bell Telephone Laboratories, Claude E. Shannon publicou um dos mais notáveis artigos científicos na história da Engenharia [1]. Esse artigo deu início a uma nova era que permitiu aos engenheiros pela primeira vez quantificar a “informação”. A ideia era descrever informação de uma maneira quantificada, que então poderia nos proporcionar:

- Analisar informações características de fontes, receptores, canais etc.;
- Aperfeiçoar a transmissão de informação através de códigos de fonte para compressão;
- Melhorar a confiabilidade de um sistema de transmissão através de códigos de canais para detecção e correção de erros.

Surge então a dúvida, como medir a informação. Essa questão no ponto de vista conceitual não é tão simples e podem haver inúmeras respostas possíveis, pode-se então contentar-se com a resposta padrão introduzida por Shannon em seu artigo [1]. A definição do conteúdo informativo de uma fonte deve depender, naturalmente, das probabilidades da respectiva variável aleatória estudada em questão. Sabe-se também que quanto maior for a incerteza associada a uma fonte, maior é a quantidade de informação que é transmitida a um observador por cada amostra gerada por essa fonte demonstrado em [3]. Logo associa-se a medida de informação procurada à medida de incerteza da fonte. É comum que essa medida de incerteza receba o nome de *Entropia*, conceito bastante enfatizado por James Massey em 1980 [2]. Tendo em mãos a medida de informação e as definições de probabilidade de cada possível símbolo gerado pela fonte, pode-se realizar as três medidas citadas acima, analisar as informações, aperfeiçoar a transmissão e melhorar a confiabilidade do sistema. O sistema desenvolvido é manipulado com o tratamento de imagens passando por um sistema de comunicação, desde sua geração até a recepção, passando por uma codificação de fonte, codificação de canal, adição de ruído, decodificação de canal e decodificação de fonte.

2 Aperfeiçoando a transmissão através de códigos de fonte para compressão

De posse da análise das informações características da fonte, pode-se então utilizar um método eficaz de otimização de um sistema de comunicações, que é a compressão de dados através de uma codificação de fonte. O código

escolhido neste estudo para compressão foi o código de **Huffman** [4]. O algoritmo de *Huffman* foi proposto por *David Albert Huffman* enquanto estudante de pós-graduação no MIT como parte de uma monografia para a aula de *Robert Fano* para resolver o problema de uma codificação óptima de uma fonte sem memória.

O Algoritmo é dado:

Considera-se uma fonte X emitindo símbolos de um alfabeto com probabilidades p_i . O algoritmo de *Huffman* pode ser dividido em duas partes.

Parte A:

- **Passo 1:** Ordena-se os símbolos por ordem decrescente de probabilidade;
- **Passo 2:** Agrupa-se os dois símbolos menos prováveis em um único símbolo cuja probabilidade será a soma das probabilidades dos símbolos agrupados (o alfabeto agora possui um símbolo a menos);
- **Passo 3:** Se o alfabeto resultante possuir dois ou mais símbolos, deve-se retornar ao passo 1, caso contrário, fim da parte A.

Parte B: A parte A do algoritmo produziu uma árvore binária na qual as folhas correspondem aos símbolos da fonte. Agora deve-se percorrer a árvore da raiz até a folha atribuindo, arbitrariamente, os símbolos “0” e “1” a cada par de ramos com origem em cada nó interno da árvore.

A figura abaixo demonstra a construção da árvore binária pelo código de *Huffman*, em que $m0, m1, m2, m3$ e $m4$ são símbolos gerados pela fonte, reorganizados de cima para baixo em forma decrescente, somando-se os dois valores menores, reorganizando novamente até que reste apenas dois símbolos.

A, B, C e D são resultados das somas demonstradas na imagem.

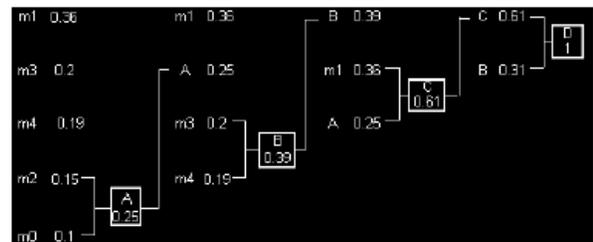


Fig 1. Árvore desenvolvida na codificação de *Huffman*.

Através desse algoritmo de codificação de fonte, podemos comprimir nossa informação de forma eficaz, demonstrando, através do *software* desenvolvido neste estudo, a compressão de uma imagem através do código de *Huffman* para aperfeiçoar a sua passagem pela linha de transmissão, demonstrado matematicamente em [4], podendo ser visualizado na tela do *software* ilustrado na Fig. 2:

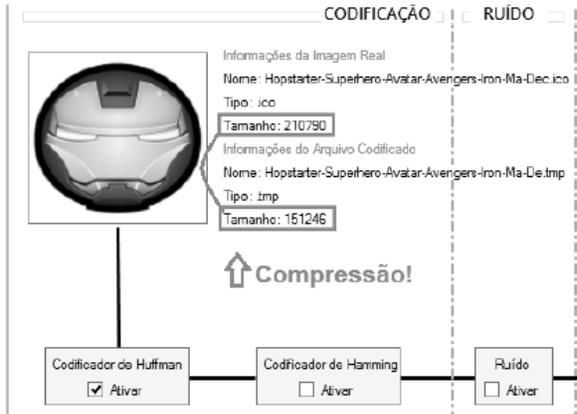


Fig 2. Compressão por *Huffman*.

Nota-se uma variação de tamanho de 210.790 bytes para 151.246 bytes, numa compressão de aproximadamente 28,248% da imagem original.

Observa-se então, uma eficiência na utilização do código de *Huffman* comprovado com o uso do *software*, que demonstra, na prática, de forma didática e interessante, a utilização do código de *Huffman* para compressão e otimização da transmissão.

3 Melhorando a confiabilidade de um sistema de transmissão através de códigos de canais para detecção e correção de erros

Com o arquivo comprimido em mãos, pode-se utilizar um sistema para tentar minimizar os efeitos de interferências e ruídos externos. O tratamento que deve-se ter para evitar ou tentar minimizar esse ruído foi a codificação de canal com detecção e correção de erros através do algoritmo do código de **Hamming**. Esse código foi desenvolvido por *Richard Wesley Hamming*, um matemático americano em seu artigo *Error Detecting and Error Correcting Codes* [5], em 1950. Ele é um código de bloco linear e sua utilização permite a transferência e armazenamento de dados de forma segura e eficiente. O código de *Hamming* utilizado no processo no *software* foi o

código de Hamming (7,4) que pode detectar até dois erros por bloco e corrigir até um erro por bloco.

Este tipo de código de controle de erros, transforma cada bloco de 4 bits de dados num bloco de 7 bits, acrescentando 3 bits de paridade.

Foram seguidas as seguintes etapas:

Construção de G e H:

A matriz **G** é denominada matriz geradora de um código linear, e **H** é chamada de matriz de paridade, tendo em vista que $\mathbf{H} \cdot \mathbf{G}^T = \mathbf{0}$ [5].

A matriz geradora e a matriz de paridade são:

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{H} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Fig 3. Matrizes Geradora e De Paridade.

Codificação de Canal:

Dado as matrizes acima, temos $k = 4$ (em que k representa o comprimento da mensagem) e $2k = 16$ palavras-código. As palavras código deste código binário podem ser obtidas a partir de .

Sendo: , em que $a_1a_2a_3a_4$ e a_i corresponde à mensagem original a ser transmitida, no campo de Galois(2) (corpo finito com dois elementos).

Pode-se analisar o *software* já com o codificador e decodificador acionados no sistema de transmissão. Observa-se ainda que há uma melhora significativa à qualidade da imagem no receptor, tendo em vista a inserção de ruído, quando se aciona o codificador e decodificador de *Hamming*, nota-se uma melhora na qualidade da imagem, já quando ele está desativado, a imagem é mais corrompida pela ação do ruído inserido.

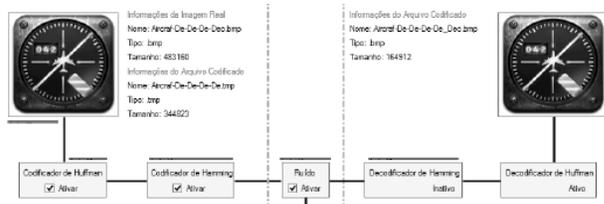


Fig 4. Eficiência do código de Hamming.

Nota-se que a imagem codificada (direita) é bem semelhante à imagem original (esquerda).

4 Inserção do Ruído Aditivo Gaussiano Branco

Tendo feito todo o processo de codificação de fonte (compressão) e de canal, transmite-se a informação através de um canal de comunicações que irá inserir ruídos na mensagem original. O ruído que foi inserido para via de simulação e demonstração foi o *Ruído Aditivo Gaussiano Branco* (AWGN – *Additive White Gaussian Noise*) [6], que é descrito a seguir.

O Ruído Aditivo Gaussiano Branco é um modelo de ruído básico utilizado para simular o efeito de processos aleatórios que ocorrem na natureza. Suas características são:

- Aditivo: Ele é adicionado a qualquer ruído que possa ser intrínseco ao sistema de informação;
- Gaussiano: Ele tem uma distribuição normal no domínio do tempo, com valor médio igual a zero;
- Branco: Refere-se ao poder que ele tem de ser aplicado a qualquer faixa de frequência para o sistema de informação. É uma analogia à coir branca que tem distribuição uniforme em todas as frequências no espectro visível.

Ao inserir-se o ruído, sem passar por nenhum processo de codificação ou compressão, obtêm-se o seguinte resultado:

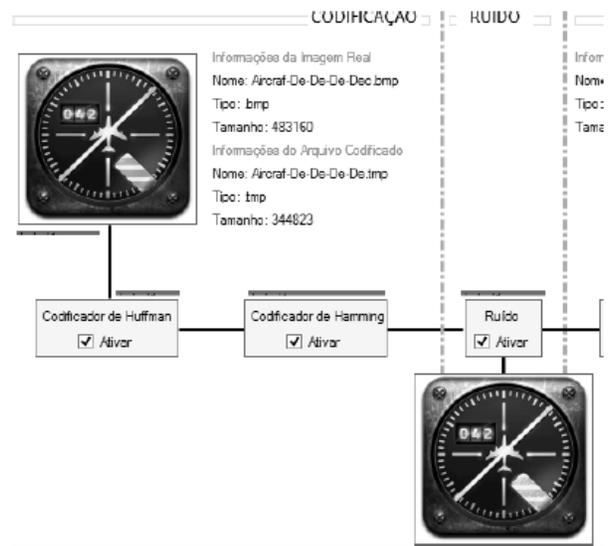


Fig 5. Inserção de ruído.

Percebe-se uma diferença de qualidade entre a imagem original (esquerda) e a corrompida (direita).

E o objetivo do ruído é, justamente, deformar uma imagem, de forma que ela se torne imperceptível ou quase imperceptível, possuindo muitas falhas quando chega ao receptor.

5 Decodificação de Cana

Após passado pelo canal, a informação precisa ser decodificada, e se pode utilizar-se de uma *decodificação computacional* descrita em [7]. Para isto, para o código de *Hamming(7,4)*, podemos definir a matriz de verificação $(n-k)$ por n , H ,

$$H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Suponha agora que o código binário recebido seja R e a mensagem transmitida seja C , determinamos o vetor erro como $E = R - C$. Calculamos então o produto do vetor recebido e a matriz de verificação de paridade

$$\begin{aligned} S &= R \cdot H \\ &= (C + E) \cdot H \\ &= C \cdot H + E \cdot H \end{aligned}$$

Como C é a palavra-código, ela satisfaz a equação de verificação de paridade $C \cdot H = 0$. Portanto,

$$S = E \cdot H.$$

O vetor S de dimensão $(n-k)$ é chamado de *síndrome de erro*. Podemos utilizar a síndrome para determinar a posição de qualquer erro único de bit. A síndrome de $[0, 0, 0]$ significa que nenhum erro ocorreu.

Podemos verificar quantos erros ocorreram e qual a posição deles através da *Tabela de Síndromes*:

Tabela 1. Tabela de Síndrome

| Líder | Síndrome |
|----------|----------|
| 00000001 | 000001 |
| 00000010 | 000010 |
| 00000100 | 000100 |
| 00001000 | 001000 |
| 00010000 | 010000 |
| 00100000 | 100000 |
| 01000000 | 101011 |
| 10000000 | 010111 |

De posse da tabela, podemos decodificar a mensagem de forma eficaz.

A decodificação de *Huffman* se dá através da árvore binária gerada pelo código de *Huffman*, obtendo assim, a imagem no tamanho original.

6 Resultados

Observa-se na imagem a seguir, que ao acionarmos todos os elementos do nosso sistema, i.e., o compressor de *Huffman*, o codificador de *Hamming* e o Ruído Aditivo Gaussiano Branco, decodificando e voltando a imagem ao tamanho original, obtêm-se uma grande melhora no tratamento da imagem, i.e., obtêm-se resultados melhores em termos de qualidade de visualização, permitindo, assim, que o usuário do *software* visualize a otimização no sistema de comunicações ao utilizar os processos de segurança da informação, protegendo-a de ações ruidosas e evitando maiores perdas. Isso demonstra a eficiência de códigos construídos e utilizados em Comunicações.

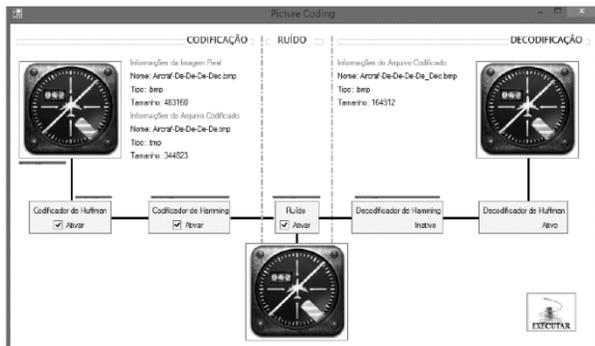


Fig 6. Interface do Software.

7 Propostas

Basicamente, esse *software* é composto por um sistema de comunicações digitais e, no momento, faz uso de três algoritmos: Um para codificação de fonte, um para codificação de canal e um para inserção de ruído. Entretanto, o objetivo futuro é tornar o software mais robusto, incluindo alguns outros algoritmos, proporcionando opções e variedade de utilização para o usuário e para o ensino da Engenharia. A proposta é de, além da inclusão de outros codificadores de fonte e canais e outros tipos de ruído, utilizar também modulação de diversos tipos e entrelaçadores diversos, para otimizar ainda mais o sistema de comunicações adotado pelo software

8 Agradecimentos

Os autores agradecem a colaboração financeira e estrutura da Escola Politécnica de Pernambuco da Universidade de Pernambuco. Agradecem também a colaboração moral e intelectual dos membros integrantes do Grupo de Pesquisa em Telecomunicações (GCom), em especial ao Igor Menezes e à Mariana Cassundé pelo apoio intelectual e materiais de apoio para a construção do *software*, ao Professor Ph.D. Valdemar Rocha da UFPE pelo auxílio no desenvolvimento científico e do conhecimento e a Antônio Nelson pelo auxílio referente à programação e a solução de problemas e barreiras encontradas.

Referências

- [1] C. E. Shannon, “A Mathematical Theory of Communications”, Bell System Technical Journal, v. 27, pp. 379-423, 623-656, Julho e Outubro de 1948.
- [2] J. L. Massey, “Applied Digital Information Theory I, Lecture Notes”, Swiss Federal Institute of Technology (ETH), Zurique, Suíça, 1980.
- [3] R. Togneri, “Information Theory and Coding”, The University of Western Australia, Janeiro de 1999.
- [4] T. M. Cover & J. A. Thomas, “Elements of Information Theory”, 2ª Ed., John Wiley and Sons, Inc., 2006.
- [5] R. W. Hamming, “Error Detecting and Error Correcting Codes”, Bell System Technical Journal, v. 29, n. 2, pp. 147-160, Abril, 1950.
- [6] A. E. Gamal, M. Mohseni and S. Zahedi, “On Reliable Communication over Additive White Gaussian Noise Relay Channels”, Information

Systems Lab, Department of Electrical Engineering - Stanford University, Stanford, CA, 2005.

- [7] S. Haykin, M. Moher, "An Introduction to Digital and Analog Communication", 2nd Edition, John Wiley and Sons, Inc., 2007.