Dissertação de Mestrado

Modelo de Minimização de Entropia da informação Aplicado à Compressão de Dados Digitais

Marcos Antonio Barbosa Lima
mabl@ic.ufal.br

Orientador:

Prof. Dr. Leandro Melo de Sales

Maceió, Setembro, 2022

Marcos Antonio Barbosa Lima

Modelo de Minimização de Entropia da informação Aplicado à Compressão de Dados Digitais

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência de Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Prof. Dr. Leandro Melo de Sales

Maceió, Setembro, 2022

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

L732m Lima, Marcos Antonio Barbosa.
Modelo de minimização de entropia da informação aplicado à compressão de dados digitais / Marcos Antonio Barbosa Lima. – 2022.
83 f. : il.

Orientador: Leandro Melo de Sales.
Dissertação (mestrado em Ciência da Computação) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.

Bibliografia: f. 80-83.

1. Entropia. 2. Informação. 3. Compressão de dados (Computação). 4. Algoritmos. I. Título.

CDU: 519.722

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PPGI
Instituto de Computação/UFAL
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401

Folha de Aprovação

MARCOS ANTONIO BARBOSA LIMA

MODELO DE MINIMIZAÇÃO DE ENTROPIA DA INFORMAÇÃO APLICADO À COMPRESSÃO DE DADOS DIGITAIS

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 31 de agosto de 2022.

Banca Examinadora:

Documento assinado digitalmente
gov.br LEANDRO MELO DE SALES
Data: 03/10/2022 18:43:38-0300
Verifique em https://verificador.iti.br

Prof. Dr. LEANDRO MELO DE SALES
UFAL – Instituto de Computação
Orientador

Documento assinado digitalmente
gov.br EVANDRO DE BARROS COSTA
Data: 03/10/2022 17:18:38-0300
Verifique em https://verificador.iti.br

Prof. Dr. EVANDRO DE BARROS COSTA
UFAL – Instituto de Computação
Examinador Interno

Documento assinado digitalmente
gov.br JOSEANA MACEDO FECHINE REGIS DE ARAU
Data: 30/09/2022 15:10:13-0300
Verifique em https://verificador.iti.br

Prof. Dr. JOSEANA MACEDO FECHINE
Universidade Federal de Campina Grande-UFCG
Examinador Externo

Agradecimentos

Agradeço aos meus colegas do Laboratório de Computação Científica e Visualização (LCCV), pela receptividade e apoio.

Agradeço ao meu orientador Leandro de Sales, por acreditar e incentivar. Ter sua orientação é um diferencial na vida de qualquer aluno.

Agradeço especialmente a minha família pela paciência e apoio ao longo de tantos anos, à minha esposa pela sabedoria e estabilidade, aos meus filhos pela motivação. É um grande privilégio compartilhar o grande sonho que chamamos vida.

Resumo

Neste trabalho são apresentados os resultados da avaliação experimental de um modelo proposto de minimização de entropia da informação em processos de compressão de dados digitais. Durante a pesquisa foram analisados modelos matemáticos e alfabetos utilizados em codificação de dados digitais. A abordagem adotada foi a codificação de dados digitais em estrutura geométrica totalmente simétrica no espaço. Na execução dos experimentos, foram monitorados diversos aspectos relacionados ao processo de codificação de dados, como esforço computacional, taxa de compressão, entropia inicial e final. Como resultado deste trabalho, constatou-se a eficácia do modelo de redução de entropia de dados, o qual reduz a zero a incerteza de informações em aglomerados de dados digitais. Desta forma, os processos de codificação resultantes deste modelo terão como saída uma quantidade constante de dados, independentemente do tamanho dos conjuntos de origem e permitem a decodificação sem perdas de dados, reduzindo assim a entropia da informação a níveis próximos a zero.

Palavras-Chaves: Entropia, Informação, Codificadores, Compressão de dados, Algoritmo

Abstract

In this work, the results of the experimental evaluation of a proposed model of information entropy minimization in digital data compression processes are presented. During the research, mathematical models and alphabets used in encoding digital data were analyzed. The approach adopted was the encoding of digital data in a geometric structure totally symmetric in space. During the execution of the experiments, several aspects related to the data encoding process were monitored, such as computational effort, compression rate, initial and final entropy. As a result of this work, the effectiveness of the data entropy reduction model was verified, which reduces the uncertainty of information in digital data clusters to zero. In this way, the encoding processes resulting from this model will output a constant amount of data, regardless of the size of the source sets and allow lossless decoding of data, thus reducing the entropy of information to levels close to zero.

Keywords: Entropy, Information, Encoders, Data Compression, Algorithm.

Conteúdo

Lista de Figuras

Lista de Tabelas

1

Introdução

É amplamente esperado que as tecnologias digitais conectadas à Internet desempenhem um papel fundamental nas atividades humanas, ajudando no desenvolvimento da sociedade em seus diferentes espectros. No entanto, o crescimento do número de dispositivos conectados, tipo de serviços e os níveis de tráfego, processamento e armazenamento de dados, impactam sensivelmente na qualidade do acesso a estes serviços.

De acordo com Salahuddin and Alam (2016), a proporção de usuários da Internet aumentou constantemente para mais de 90% em muitos países economicamente desenvolvidos e à medida que as infraestruturas digitais, serviços e produtos que suportam se expandem cada vez mais, mesmo em países onde o acesso à Internet já é generalizado, as implicações e necessidade de infraestrutura também crescem proporcionalmente, segundo Lange et al. (2020), isto pressiona qualitativamente esses serviços. De acordo com Lange et al. (2020), o crescimento do número de usuários da Internet varia de 20% a cada três anos e o tráfego na Internet tende a dobrar a cada ano, mesmo em relatórios governamentais mais conservadores. Em Granger et al. (1998), observa-se que a Internet se tornou uma parte essencial da vida moderna, onde 4,66 bilhões de pessoas acessam diariamente páginas interativas, com conteúdo multimídia de todos os tipos .Para se ter uma ideia, conforme discutido em , em 2020 a quantidade de dados armazenada no universo digital era de 44 Zettabytes, um número impressionante de 10^{21} bytes sendo, em média, criados 1,7 MB de dados por segundo, por pessoa, com acréscimo de 39 milhões de novos usuários naquele ano. Além disso, o crescimento da quantidade de dados na rede previsto para os próximos cinco anos, permite-se estimar em 200 Zettabytes de dados armazenados em 2025 (Figura 1.1).

Um vídeo de baixa resolução (por exemplo, 480 *pixels* / *frame*) transmitido em uma conexão de *streaming* entre um servidor e um dispositivo conectado, consome, em média, 500 MB por hora, ao passo que um vídeo de alta qualidade (por exemplo, 4K - 8.294.400 *pixels* / *frame*) consome, em média, 6 vezes mais que o de baixa qualidade. Este é um exemplo, dentre vários, que claramente demonstra o aumento na demanda por trânsito de dados a níveis gigantescos

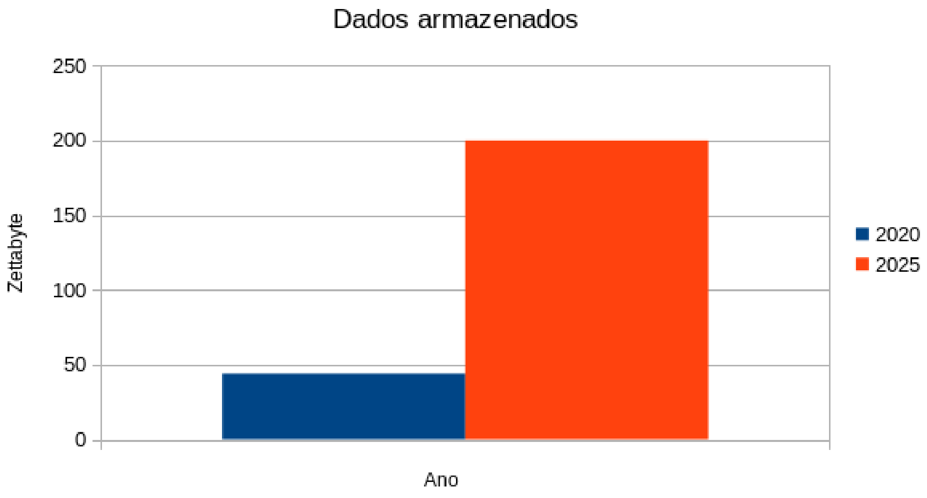


Figura 1.1: Expansão dos dados armazenados até 2025 Lange et al. (2020).

(Figura 1.2), comparando-se a alguns anos atrás. Neste contexto, várias linhas de pesquisas tecnológicas estão em desenvolvimento, visando aumentar a disponibilidade e qualidade dos serviços de trânsito de dados na Internet.

Figura 1.2: Utilização de banda passante por hora Aguiar and Martens (2016).

1.1 Problemática

De acordo com Nguyen and Jaumard (2009), para atendimento às demandas de crescimento da base de armazenamento e trânsito de dados na Internet, o desenvolvimento de novas tecnologias tem sido voltado a dispositivos eletrônicos de retransmissão de dados, cujas gerações apresentam aumento de frequência de processamento, tabelas de alocação, núcleos de processamento e software .

A dinâmica de desenvolvimento de novas gerações está estabelecida e têm atendido a demanda de crescimento, porém o crescimento da capilaridade da Internet pressiona a necessidade de aumento de novas instalações físicas (*data centers*) com os respectivos custos. Além disso, a estagnação do crescimento da frequência de processamento pode impactar neste ciclo de evolução, criando bloqueios ao desenvolvimento orgânico e operação da Internet.

A compressão de dados apresenta-se como uma possível solução às necessidades descritas, uma vez que permite a redução do tráfego de dados e, consequentemente, o crescimento da rede sem bloqueio ou gargalos. Entretanto, as tecnologias de compressão de dados sem perdas apresentam uma limitação quanto ao limite que esta compressão pode ocorrer. Em geral, os ganhos obtidos são decorrência da redução ou eliminação da redundância em conjuntos de dados e este processo apresenta um limite estabelecido pela entropia deste conjunto de dados, não podendo haver ganhos superiores a esta entropia. Sendo os básicos ou avançados, bem como funcionais em produção, atualmente utilizam-se diferentes modelos de compressão de dados que são limitados por uma entropia H média gerada por uma fonte (mais detalhes no Capítulo 2).

Diante das limitações já conhecidas das atuais abordagens de compressão de dados, o escopo deste trabalho foi a busca por um modelo de codificação de dados que permita a redução da entropia da informação e a expansão da capacidade de compressão de dados a níveis maiores dos encontrados atualmente.

1.2 Delimitação do Trabalho

Diversas abordagens e linhas de pesquisa buscam otimizar a taxa de compressão, direcionando seus esforços a grupos de aplicações de conjuntos de dados. Sendo assim, técnicas como:

1. Transformada de Burrows–Wheeler [Li and Durbin (2009)];
2. Modelagem probabilística [Legay et al. (2010)];
3. Codificação Aritmética [Wallace (1992)];
4. Transformada discreta de cosseno (DCT) [Narasimha and Peterson (1978)];

dentre outras, atendem às demandas de áreas específicas. Neste trabalho, o foco é em compressão sem perdas, e neste contexto, as técnicas mais utilizadas podem ser divididas em três grupos, conforme Salomon (2002):

1. Métodos de redução de entropia;
2. Métodos baseados em dicionários;

3. Transformações.

Os algoritmos Huffman Connell (1973), LZW (Lempel–Ziv–Welch) Ziv and Lempel (1978) e Burrows-Wheeler Li and Durbin (2009), apesar das diferentes abordagens, em geral tem como objetivo a eliminação da redundância de blocos de dados e consequente redução da variância de ocorrências destes blocos. O mesmo acontece para o método de Burrows-Wheeler que, apesar da classificação no grupo de transformação, apresenta como diferencial uma análise estatística da probabilidade de ocorrência de blocos e seus posteriores.

Para efeito de análise neste trabalho, será utilizado o algoritmo LZW, uma vez que sua implementação, execução e complexidade são de baixa exigência computacional e os resultados apresentam pouca variação em relação aos demais.

1.3 Relevância

Uma das linhas de pesquisa proeminentes é a compressão de dados baseada em abordagens matemáticas, apresentando-se como um possível caminho para mitigar os gargalos tecnológicos diante do aumento constante da demanda. No passado, este tema foi muito pesquisado, mas com o tempo se atingiu um equilíbrio científico, não observando-se tantas pesquisas nesse sentido nos últimos tempos. Em todo caso, este trabalho revisita diversos aspectos de abordagens matemáticas conhecidas a fim de buscar reduzir a quantidade de dados em trânsito e melhorar os serviços de Internet, ao passo que aumentar o nível de escalabilidade no ponto de vista de usuários conectados às redes.

De acordo com Lange et al. (2020), o crescimento da demanda pelos serviços baseados na Internet em diferentes áreas coloca as redes e as infraestruturas de armazenamento como centro de informações e negócios da sociedade, sendo agente de progresso de várias nações. Diferentes áreas e serviços têm pressionado este desenvolvimento, uma vez que os dados gerados tem um crescimento maior que a capacidade da infraestrutura instalada, criando entraves a este desenvolvimento. Algumas áreas em particular podem ser beneficiadas com o avanços das pesquisas nesse contexto, tais como:

1. **Arquivos de Imagem:** Os dispositivos geradores de imagem evoluíram em qualidade muito rapidamente, isto implica em maior quantidade de *pixels* e por consequência um crescimento da geração de dados em ordem de grandeza cada vez maior. O padrão 480 *pixels* em uma conexão de *streaming* transita entre o servidor e o dispositivo conectado em média 500 MB por hora. A disponibilização de cada vez mais dispositivos e sua facilidade de uso proporciona um salto na criação e uso de dados na Internet.

2. **Arquivos de Áudio:** A mesma dinâmica de imagem ocorre para arquivos de áudio. Notadamente, o uso de música pela Internet tornou-se padrão e, neste caso, a demanda é maior por trânsito que por geração de dados.

3. **Cloud Computing:** A tendência de transferência de ativos de empresas para a Internet tornou-se padrão, onde os serviços de fornecidos estruturas próprias tendem a desaparecer em pequenas e médias empresas. Todo o legado de dados será transferido e a geração de novos dados agora fará parte do eco sistema da Internet, sendo este um fator de pressão e criticidade de serviços muito expressivo.

4. **Serviços de entretenimento:** O popularização de serviços de entretenimento por *streaming* como jogos e vídeos são fatores de grande pressão por crescimento da infraestrutura

5. **Internet da coisas:** A tecnologia chamada Internet das Coisas aumenta a capacidade do homem e dos computadores de controlar bilhões de dispositivos de conectividade. A utilização da IoT como um sistema proporcionará mudanças da interação humana com o mundo Neeli and Patil (2021) e será um grande fator de crescimento da demanda de fluxo de dados na Internet.

6. **Aumento da capilaridade de uso:** Em 2020, 39 milhões de novos usuários passaram a ter acesso à Internet Lange et al. (2020). Com um taxa de crescimento de 20% da quantidade usuários a cada três anos, a capilarização representa um fator de crescimento multiplicador dos fatores já discutidos.

1.4 Objetivos do Trabalho

Com base nas discussões apresentadas, o principal objetivo neste trabalho é apresentar e analisar um novo modelo de minimização de entropia em conteúdos digitais, baseado em uma abordagem geométrica e iterativa, que possibilita um processo de codificação e decodificação independente da entropia, sendo um método determinístico de compressão, com saída fixa e independente do tamanho da fonte de dados.

1.4.1 Objetivos Específicos

1. Revisitar as abordagens mais tradicionais e modernas sobre compressão de dados;

2. Descrever e discutir o modelo de minimização de entropia aqui denominado MME;

3. Avaliar a efetividade e o desempenho do modelo de minimização de entropia da informação em dados digitais, considerando-se sequências aleatórias de dados e independente de formato do arquivo;

4. Apresentar os resultados de compressão utilizando o MME em comparação ao algoritmo de compressão LZW (Lempel–Ziv–Welch) Ziv and Lempel (1978), considerando-se as

métricas entropia e após a aplicação dos processo de compressão, bem como os discussões sobre os ganhos obtidos com o uso do MME.

1.5 Estrutura da Documento

Neste documento, os capítulos foram organizados de forma a fornecer a base teórica dos métodos de compressão e suas limitações, bem como o método proposto e os resultados, de acordo com a seguinte estrutura:

- No Capítulo 2, apresentam-se os principais conceitos da Teoria da Informação, Entropia e Compressão de dados.
- No Capítulo 3, apresentam-se os principais trabalhos relacionados.
- No Capítulo 4, apresenta-se o Modelo de Minimização de Entropia (MME) da informação em dados digitais, no qual é discutida a estrutura geométrica para codificação e decodificação.
- No Capítulo 5, apresentam-se os resultados dos experimentos de compressão de dados utilizando o modelo proposto e discussões comparativas com o algoritmo LZW (Lempel–Ziv–Welch).
- No Capítulo 6, apresentam-se as conclusões e trabalhos futuros.

2

Entropia da Informação

Neste capítulo são apresentados os conceitos diretamente ligados à pesquisa do modelo de minimização de entropia estudado: Entropia, Codificação e Compressão de dados. Inicialmente, será tratada a teoria da informação e suas conclusões referentes à quantificação da entropia, seus valores condicionais e redundâncias de informação. Em seguida, serão estudadas as abordagens de compressão de dados e tipos de codificadores. O principal objetivo aqui é entender que os métodos descritos a seguir possibilitam quantificar os bits mínimos à compressão de um conjunto de dados, que aplicados às técnicas de codificação descritas, originam alguns dos sistemas atuais de compressão de dados sem perdas.

2.1 Teoria da Informação e Entropia

Em 1948, Shannon (1948) lançou as bases da Teoria da Informação, com base no problema enfatizado sobre como reproduzir uma mensagem exata ou a mais próxima possível da mensagem originalmente emitida, quantificando-a através de uma grandeza e sem ambiguidades.

De acordo com Pineda et al. (2006) , a quantidade de informação atribuída a um sistema está ligada necessariamente à variação de símbolos emitidos para a formação de uma mensagem, ou seja, quanto maior a variação de símbolos, maior será a quantidade de incerteza e, por consequência, de informação ligada a fonte . Para exemplificar o conceito de quantidade de informação e incerteza na Teoria da Informação, pode-se usar como exemplo um jogo de dados, onde se tem seis possibilidades ou seis estados acessíveis, tendo uma incerteza probabilística a quantidade de informação. Nesta base de Shannon (1948), faz-se uso de uma função logarítmica, tendo sua forma representada na Equação 2.1.

$$H(p_1, ..., p_n) = -\sum_{i=1}^{n} p_i \log(p_i) \tag{2.1}$$

em que p_i é a probabilidade de ocorrência de cada evento i em um conjunto de amostras n e $\sum_{i=1}^{n} p_i = 1$. Nesse tratamento, um certo evento i representa um certo símbolo emitido pela fonte. Nesse caso, a Equação 2.1 representa a entropia de Boltzmann-Gibbs, ou seja, trata estatisticamente uma mensagem levando em consideração os símbolos que a compõem, com seus múltiplos constituintes. Assim, o termo entropia ganha um novo significado, o de indicar a medida de incerteza probabilística em uma dada distribuição de probabilidade, passando a ser denominada entropia de *Shannon*. A noção de entropia possui características que servem de base a Teoria da Informação:

1. A entropia máxima é atingida quando a ocorrência de todos os símbolos é equiprovável, não existindo tendência de concentração de probabilidades em algum símbolo.

2. A medida que a ocorrência de um símbolo se torna mais provável que a dos outros símbolos do repertório, a entropia decresce.

3. Quando existe certeza sobre qual símbolo vai ser transmitido, a entropia é zero.

A incerteza probabilística está ligada à quantidade total n dos possíveis eventos i. Assim, tem-se que H cresce monotonicamente com o aumento do número total n de possibilidades i que o sistema possui. Com infinitas possibilidades, tem-se uma incerteza probabilística infinita, assim H passa a ser escrita como H = H(n).

Existem circunstâncias onde os eventos tem probabilidades de ocorrência diferenciadas. Por exemplo, no envio de uma mensagem, uma distribuição de símbolos não deve ser equiprovável, uma vez que uma mensagem minimamente inteligível necessita de concentração de agrupamentos de símbolos. Assim, tem-se que a medida de incerteza H também depende da probabilidade p_i de ocorrência de cada símbolo i. Para cada símbolo i existe uma medida de incerteza I que depende da sua probabilidade de ocorrência p_i. Essa medida I(p_i) é chamada de auto-informação. A quantidade de informação média, ou incerteza probabilística, associada a um conjunto de eventos possíveis, será então a média das auto-informações ponderadas pela probabilidade da ocorrência de cada evento i, em particular definida na Equação 2.2.

$$H(p_1, ..., p_n) = -\sum_{i=1}^{n} p_i I(p_i) \tag{2.2}$$

Conforme definida por do Lago Mendes (2017), a auto-informação I(p_i) de cada evento i é representada por uma função logarítmica, com as seguintes propriedades:

1. A quantidade de informação possui uma propriedade aditiva;

2. A quantidade de informação I(p_i) é máxima, quando os i eventos tem a mesma probabilidade de ocorrência;

3. Quando a escolha é subdividida em duas sucessivas, a informação original do conjunto deve ser a soma ponderada das individuais.

Desta forma, a auto-informação I(p_i) do evento i assume o formato definido na Equação 2.3:

$$I(p_i) = -\log(p_i) \tag{2.3}$$

Deve ser observado que p_i é um número menor que 1.0 e, como a expressão é negativa, a auto-informação do evento é tanto maior quanto menor for a probabilidade de sua ocorrência. Assim, substituindo a Equação 2.2 na Equação 2.3, obtém-se a Equação 2.4.

$$H(p_1,...,p_n) = -\sum_{i=1}^{n} p_i \log(p_i) \tag{2.4}$$

Para a entropia máxima, ou seja, quando as possibilidades são equiprováveis, a Equação 2.3 toma a seguinte forma:

$$H = \log n \tag{2.5}$$

A base do logaritmo utilizada até aqui é arbitrária, Shannon em seu trabalho escolheu a base 2, o que define o bit, a unidade de medida na Teoria da informação e a função que a representa como uma função côncava. A sua visualização é definida tendo uma variável aleatória com dois valores possíveis, conforme definição na Equação 2.6.

$$H(p) = -p\log p - (1-p)\log(1-p) \tag{2.6}$$

2.2 Entropia de Shannon Relativa

Considerando X e Y duas variáveis aleatórias com as respectivas distribuições $p_i = (p_1,...,p_n)$ e $q_i = (q_1,...,q_n)$, conforme Yanagi et al. (2005), pode-se definir a entropia relativa da variável X em relação a variável Y, conforme Equação 2.7.

$$H(X||Y) = -\sum_{i=1}^{n} p_i log(\frac{q_i}{p_i}) \tag{2.7}$$

A entropia relativa quantifica uma certa distinção entre duas distribuições de probabilidades p_i e q_i. Assim, a Entropia de Shannon Condicional quantifica o valor da incerteza probabilística do par H(X, Y) e é definida como sendo a medida da incerteza probabilística sobre a variável X, tendo a quantidade de informação referente à variável Y, conforme Equação 2.8.

$$H(X|Y) = H(X,Y) - H(Y) \tag{2.8}$$

A entropia conjunta do par de variáveis aleatórias X e Y, é definida segundo a Equação 2.9.

$$H(X||Y) = -\sum_{ij} p_{ij} log(p_{ij}) \tag{2.9}$$

em que, p_{ij} é a distribuição conjunta de (X, Y). A entropia conjunta nos fornece uma medida da incerteza do par (X, Y).

3

Trabalhos Relacionados

3.1 Compressão de Dados

O objetivo principal da codificação é representação de um conjunto de dados usando um número de bits tão pequeno quanto possível, preservando a integridade das informações contidas deste conjunto. A codificação de dados possui duas principais aplicações: a redução da utilização banda passante requerida por sistemas de transmissão de dados digitais, como *streaming*, vídeo conferências etc. e a redução dos insumos necessários ao armazenamento como redução do hardware necessário ao armazenamento de dados de imagens usadas em vídeos digitais, bancos de dados etc. A exigência de integridade dos dados codificados em sua decodificação sofre variação de acordo com a sua utilização. Em codificação de imagens é primordial a preservação da informação da imagem, mas informações menos relevantes podem ser descartadas. Técnicas de codificação que provocam pequenas perdas de qualidade da imagem original são chamadas compressão com perdas de informação. Em processos onde a recuperação de todos os dados é imperativa, utilizam-se técnicas de compressão sem perdas de informação. O desafio de comprimir dados digitais tem sido abordado ao longo das últimas décadas com base principalmente na Teoria Algorítmica da Informação Solomonoff (1960) e na Taxa de Distorção da Teoria de Perda Blau and Michaeli (2019).

3.2 Técnicas de Compressão de Dados

Um grande número de técnicas de compressão de dados foi desenvolvido, e desta forma surge a necessidade de selecionar o algoritmo adequado a uma situação particular. Em Jayasankar et al. (2021), foram classificadas as técnicas em três categorias, a saber, reconstrução de dados, esquemas de codificação e tipo de dados, conforme Figura 3.1.

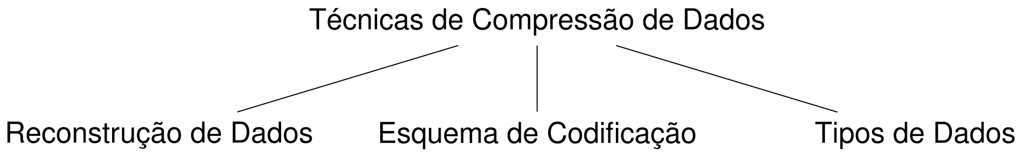


Figura 3.1: Tipos de Codificação adaptado Jayasankar et al. (2021).

3.2.1 Compressão e Reconstrução dos dados

Quando uma técnica de compressão de dados é usada para fins gerais, como mensagens ou navegação na Internet, a qualidade dos dados reconstruídos não é considerada. Mas em compressão de texto, a modificação de um único caractere em um documento não é aceitável, pois altera todo o seu significado.

3.2.2 Compressão sem perdas

Em imagens médicas ou compressão de imagens de sensoriamento remoto, pequenas alterações nos valores de pixel não são desejáveis. O significado da qualidade dos dados depende do tipo de aplicação envolvida. O requisito qualidade dos dados reconstruídos, define o tipo de compressão em compressão sem perdas e com perdas. Como o nome indica, compressão sem perdas, refere-se a nenhuma perda de informação, ou seja, os dados reconstruídos são idênticos aos dados originais. É usado em aplicações onde a perda de informação é indesejável como texto, imagens médicas, imagens de satélite, etc. (Drost and Bourbakis (2001)).

3.2.3 Compressão com perdas

Em alguns cenários, técnicas de compressão com perdas são preferíveis onde os dados reconstruídos não são perfeitamente iguais, mas aceitáveis, em relação aos dados originais. Isto possibilita uma grande vantagem para técnicas de compressão com perdas quando comparadas às técnicas de compressão sem perdas, e permite um capacidade muito maior de compressão, conforme Figura 3.2.

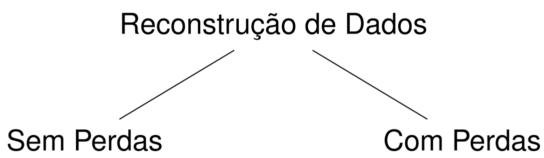


Figura 3.2: Reconstrução de Dados adaptado Jayasankar et al. (2021).

3.2.4 Compressão com Esquemas de Codificação

A compressão de dados por codificação apresenta subdivisões conforme Figura 3.3. A codificação de Huffman (Connell (1973)) é uma técnca de codificação que efetivamente compacta dados em quase todos os formatos de arquivo. É um tipo de código de prefixo ótimo que é amplamente empregado em compressão sem perdas. A ideia básica é atribuir códigos de comprimento variável para caracteres, dependendo da frequência de ocorrência. A saída é uma tabela de código de comprimento variável. A codificação de Huffman ainda é popular devido à sua implementação ser simples e sua compactação ser mais rápida. Existem várias versões do código de Huffman como:

1. Código Huffman de variação mínima;
2. Código Huffman limitado por comprimento;
3. Código Huffman não binário;
4. Código Huffman adaptativo;
5. Código Golomb;
6. Código Rice;
7. Código Tunstall;

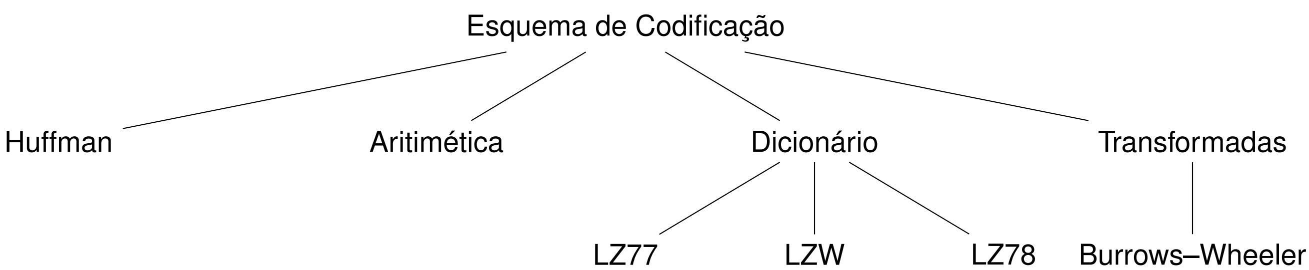


Figura 3.3: Esquemas de codificação, adaptado Jayasankar et al. (2021).

A codificação aritmética (Langdon (1984)) é uma técnica de compressão que gera códigos de comprimento variável. É superior a codificação Huffman e é muito útil em situações em que a fonte contém pequenos alfabetos com probabilidades distorcidas. Quando uma string é codificada usando codificação aritmética, símbolos que ocorrem com maior frequência são codificados com menor número de bits que símbolos que ocorrem raramente. Existem duas versões de codificação aritmética:

1. Codificação aritmética adaptativa;
2. Codificação aritmética binária.

As abordagens de codificação baseadas em dicionário são úteis em situações onde os dados originais contêm mais padrões repetidos. Quando um padrão é identificado numa sequência de entrada, ele é codificados com um índice no dicionário. Quando o padrão não está disponível no dicionário, ele é codificado e adicionado ao dicionário. O algoritmo Lempel–Ziv (LZ) é uma técnica de codificação baseada em dicionário comumente usada na compressão de arquivos sem perdas. Ele é amplamente utilizado devido à sua adaptabilidade a vários formatos de arquivo e mantém um dicionário dos padrões identificados. O comprimento do dicionário é definido a um determinado valor. Este método tende a ser mais eficaz para arquivos maiores. Nos anos de 1977 e 1978, duas versões do LZ foram desenvolvidas por Ziv e Lempel nomeadas como LZ77 (Ziv and Lempel (1977)) e LZ78 (Ziv and Lempel (1978)). Lempel–Ziv-Welch (LZW) é uma versão aprimorada do LZ77 e LZ78 que foi desenvolvida por Terry Welch em 1984 (Welch (1984)), nesta técnica, o codificador constrói um dicionário adaptativo para caracterizar as strings de comprimento variável sem conhecimento prévio da entrada. O decodificador também constrói o dicionário semelhante como codificador baseado no código recebido dinamicamente. Transformada de Burrows-Wheeler (BWT) (Blau and Michaeli (2019)) é uma técnica de compressão de classificação de blocos que reorganiza a cadeia de caracteres em execuções de caracteres idênticos. Ele usa duas técnicas para compactar dados e incluem transformação de movimentação para frente e leitura linha-a-linha e coluna-a-coluna.

3.2.5 Compressão e Tipos de dados

Quando analisadas as técnicas de compressão de dados por tipo, observam-se quatro tipos que se destacam conforme figura 3.4

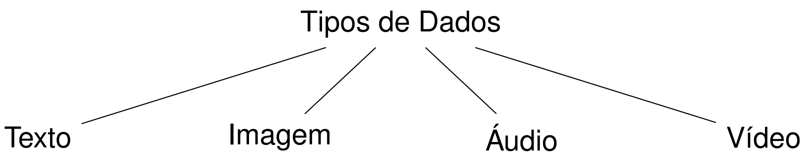


Figura 3.4: Tipos de dados Jayasankar et al. (2021).

3.2.6 Compressão de Textos

Basicamente, os dados textuais sofrem compressão sem perdas onde perda de informação não é permitida. Em (Abel and Teahan (2005)) foi criado um método universal de pré-processamento para compactar dados textuais. Ele integra cinco algoritmos que incluem conversão de letras maiúsculas, final codificação de linha, substituição de palavra, substituição de frase e gravação de alfabetos. Não tem dependência de idiomas e não precisa de dicionários. Mas, observa-se que o custo de processamento é alto.

3.2.7 Compressão de Imagem

Em Lansky and Zemlicka (2006) foi desenvolvido um novo método de compressão baseado na minimização booleana de dados binários juntamente com a codificação Burrows-Wheeler (BWT) para comprimir arquivos de texto menores.

A Codificação de Imagem Embutida mostrada em (Kailasam (2020)) é uma técnica de compressão de imagem mais simples e eficiente onde fluxos de bits são criados em ordem de importância, produzindo assim um código completamente embutido. Um código incorporado define uma sequência de decisões binárias que diferencia uma imagem de uma imagem nula ou cinza. O processo de codificação e decodificação pode ser encerrado quando a taxa alvo for atingida. O destino pode ser uma contagem de bits, quando a contagem de bits necessária for atingida, o processo de codificação será encerrado. Da mesma forma, o processo de decodificação também pode ser interrompido em um determinado ponto e reconstruir a imagem de acordo com a codificação de taxa mais baixa. A ausência de treinamento prévio e controle de taxa de precisão é a grande vantagem deste método. Outra técnica de compressão de imagem é proposta em (Rao and Eswaran (1996)), onde são apresentados dois algoritmos simples para codificação de truncamento de blocos, sendo uma maneira mais simples e rápida de implementar o algoritmo de codificação de imagem.

3.2.8 Compressão de áudio

O padrão IEEE 1857.2 desenvolvido para codificação avançada de áudio em (Auristin and Mali (2016)) é uma técnica de compressão de áudio sem perdas, projetada especificamente para obter melhor qualidade de áudio com largura de banda otimizada, além de velocidade mais alta. Envolve uma coleção de ferramentas para atingir funções específicas de codificação de áudio. Neste método, a codificação é feita por Predição Linear seguida de processamento de bloco de codificação de entropia. Esta técnica produz melhor compressão em taxas mais rápida de processo de codificação e decodificação.

Outro método de compressão de áudio, baseado no método de codificação de taxa de bits variável escalável, é proposto em (Hang et al. (2016)) que é adaptável às variações de largura de banda.

3.2.9 Compressão de vídeo

Um vídeo também é uma parte essencial dos aplicativos multimídia. Geralmente, os arquivos de vídeo consomem mais recursos para fins de comunicação, processamento e armazenamento. Portanto, a compressão é muito necessária para que os arquivos de vídeo o armazenam, processam ou transmitam. Diversas técnicas foram desenvolvidas para compactar eficientemente arquivos de vídeo para evitar que grandes quantidades de dados sejam transmitidas ou armazenadas.

O padrão H.261 foi proposto para transmitir vídeo a uma taxa de 64kbps e seus múltiplos. Os quadros do H.261 podem ser classificados em dois tipos: quadros codificados Intra (quadros I) e codificados previstos (quadros P). Nos quadros I, os quadros são codificados sem dependência dos quadros anteriores, enquanto os quadros são codificados usando quadros anteriores nos quadros P. O H.261 é idêntico ao padrão de compactação JPEG e emprega previsão temporal com compensação de movimento. É amplamente utilizado para chamadas de vídeo e videoconferência.

O H.263 é o mesmo que o H.261, mas foi especialmente projetado para taxas de bits mais baixas. A imagem é particionada em vários macro-blocos. O bloco macro possui blocos de luminância de 16×16 e blocos de crominância de 8×8. Os blocos macro são codificados como blocos intra ou inter.

Moving Pictures Experts Group (MPEG) é um grupo de trabalho ISO/IEC que desenvolve técnicas de compressão, representação de imagens em movimento e áudio em padrões internacionais descrito em (Watkinson (2012)). MPEG é um padrão de compressão de vídeo baseado em camadas que leva a um fluxo de vídeo compactado com uma taxa de bits de quase 1,5 Mbps a uma resolução de quase 352×240. Sequências de vídeo MPEG compostas de diferentes camadas que podem acessar aleatoriamente uma sequência de vídeo e proteger contra dados corrompidos. Os quadros MPEG podem ser codificados de três maneiras:

1. Os quadros I são codificados como quadros discretos que não dependem dos quadros anteriores;

2. Os quadros P são codificados para tamanhos de quadro menores do que os quadros I;

3. Os quadros B precisam de um quadro anterior e futuro (quadros P e quadros I) para fins de decodificação.

A decodificação em MPEG apresenta um esforço computacional muito alto, porém a sua escalabilidade de fluxo de bits fornece flexibilidade no poder de processamento necessário para decodificação.

3.3 Codificadores

Diversas abordagens e linhas de pesquisa buscam otimizar a taxa de compressão com seus esforços direcionados aos grupos de aplicações de conjuntos de dados. Sendo assim, técnicas como:

1. Transformada de Burrows–Wheeler Li and Durbin (2009);

2. Modelagem probabilística Legay et al. (2010);

3. Codificação Aritmética Wallace (1992);

4. Transformada discreta de cosseno (DCT) Narasimha and Peterson (1978);

dentre outras, atendem às demandas de áreas específicas, tendo abrangência limitada direcionada.

As técnicas mais utilizadas em compressão sem perdas, podem ser divididas em três grupos, conforme Salomon (2002), e ilustra-se na Figura 3.5:

1. Métodos de redução de entropia: usam as probabilidades de ocorrência dos símbolos e alteram sua representação. Assim, obtém-se a redução do número de bits usados para representar cada símbolo.

2. Métodos baseados em dicionários: buscam eliminar repetições de símbolos.

3. Transformações: buscam métricas estatísticas aplicadas as ocorrências em uma série de dados.

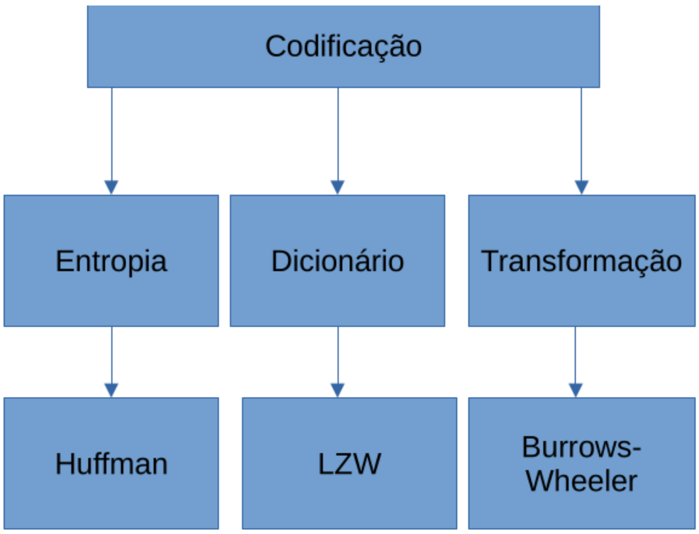


Figura 3.5: Tipos de Codificação adaptado Salomon (2002).

Uma fonte de sinais gera uma entre um conjunto de possíveis mensagens identificado como **L**, as mensagens, enumeradas como, $k = 1, ..., L$, possuem probabilidades $p_k, k = 1, ..., L$ e sua informação associada com r_k é definida conforme Equação 3.1.

$$I_k = -log_2 p_k \text{bits} \tag{3.1}$$

A Entropia (H) de uma fonte de informação é definida como sendo a informação média gerada por esta fonte Shannon (1948), conforme Equação 3.2.

$$H(X||Y) = -\sum_{k=1}^{L} p_k \log_2 p_k \tag{3.2}$$

A entropia de uma fonte fornece o número mínimo de bits necessários para codificar a informação gerada por esta fonte. De acordo com o Teorema da Codificação sem Ruído de Shannon Ma and Ma (2018), é possível codificar sem perdas a informação gerada por uma fonte com entropia de H bits, usando em média $H+\varepsilon$ bits por mensagem, onde $\varepsilon >= 0$ é uma quantidade arbitrariamente pequena. Na codificação por entropia, o objetivo é utilizar um número de bits o mais próximo possível do valor da entropia, e isto pode ser feito através de um código de comprimento variável, onde os valores com altas probabilidades são representados por palavras-código de menor tamanho. Como resultado, o conjunto de dados resultante desta codificação tenderá ao mínimo de bits necessário. O valor de ε sofrerá variação de acordo com o algoritmo utilizado.

Diversas abordagens de compressão sem perdas estão disponíveis na literatura, onde se destacam o algoritmo Huffman Connell (1973) e o LZW (Lempel–Ziv–Welch) Ziv and Lempel (1978). É interessante observar que algoritmos de codificação sem perdas utilizando palavra-código em geral são variações destas duas abordagens. Como já mencionado, neste trabalho será descrita a técnica LZW (Lempel–Ziv–Welch) para posterior comparação com o Modelo de Minimização de Entropia da informação em arquivos digitais aqui proposto.

3.4 Codificação LZW

O Algoritmo LZW (Lempel–Ziv–Welch) é resultado de melhoramentos nos algoritmos LZ77 e LZ78 Ziv and Lempel (1977), os quais foram desenvolvidos por Jacob ZIV e Abraham Lempel, respectivamente. Inicialmente, foi projetado para calcular o número de bits que se poderia reduzir em arquivos digitais e, devido a sua abrangência, tem sido utilizado em muitos outros contextos. Trata-se de um algoritmo de compressão de dados baseado em dicionário que produz códigos de comprimento fixo associados às sequências de símbolos de comprimento variável. Os códigos gerados durante o processo, tanto de codificação como de decodificação, correspondem a sequências de símbolos cada vez mais longas, que vão sendo adicionadas ao dicionário. O LZW é um algoritmo de compressão sem perdas, o que significa que o conjunto de dados, após a descompressão, é exatamente igual ao conjunto de dados originais, não se perdendo qualquer informação, conforme ilustra-se na Figura 3.6.

Para fins de exemplificação, conforme descrito por Altoé and Pinho (2005), suponha-se que durante o processo de compressão foi encontrada a sequência de símbolos *abcdefg...* e que a sequência *abcd* já se encontra na tabela de sequências, mas a sequência *abcde* ainda não se encontra na mesma. Neste caso, é armazenado no arquivo comprimido o código da sequência

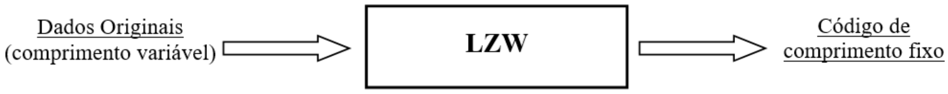
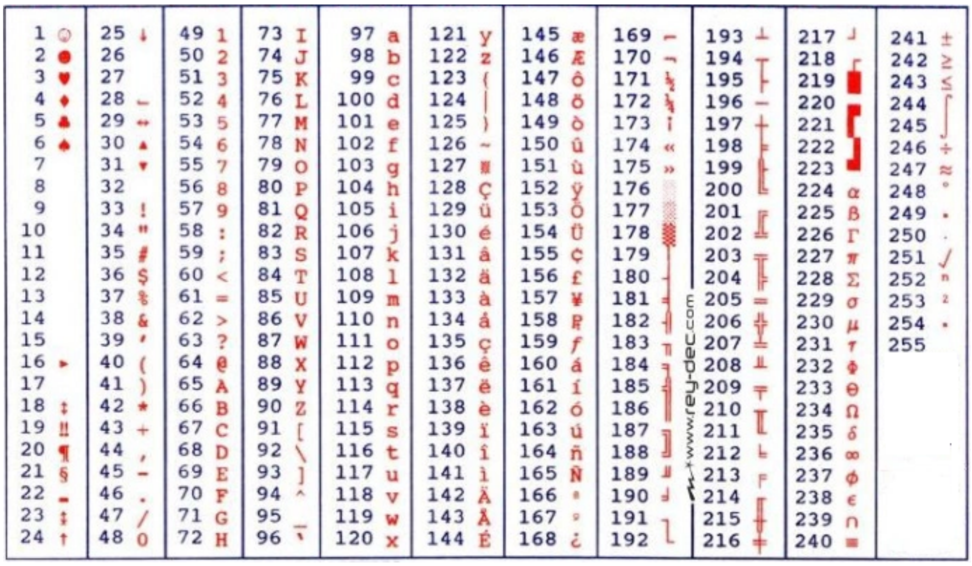


Figura 3.6: Comprimento fixo LZW (Lempel–Ziv–Welch Salomon (2002)).

abcd e a sequência *abcde* é inserida na tabela com um novo código. Continua-se o processamento com a sequência *efgh* e assim por diante. Note-se que, na primeira ocorrência da sequência *abcde*, a mesma é apenas inserida na tabela e associada a um código, mas este código não é armazenado no arquivo comprimido, o que só ocorrerá a partir da segunda ocorrência da sequência *abcde*.

Sendo assim, o processo de compressão consiste na criação de um "dicionário inicial" usando códigos de 8 ou 12 bits (atualmente, são utilizados códigos de 12 ou mais bits, porque suportam, no mínimo, 4096 registros na tabela). Por padrão, é usado o código ASCII ou o UTF8 para a criação deste dicionário, cujos caracteres ocupam as primeiras 256 linhas da tabela e as restantes serão acrescentadas com o conjunto de símbolos encontrados no arquivo durante o processo de codificação e decodificação. No contexto deste trabalho para o modelo proposto no próximo capítulo, considera-se a tabela de códigos ASCII, conforme ilustra-se na Figura 3.7.



1	☺	25	↓	49	1	73	I	97	a	121	y	145	æ	169	⌐	193	┴	217	┘	241	±
2	☻	26		50	2	74	J	98	b	122	z	146	Æ	170	¬	194	┬	218	┌	242	≥
3	♥	27		51	3	75	K	99	c	123	{	147	ô	171	½	195	├	219	█	243	≤
4	♦	28	∟	52	4	76	L	100	d	124	|	148	ö	172	¼	196	─	220	▄	244	⌠
5	♣	29	↔	53	5	77	M	101	e	125	}	149	ò	173	¡	197	┼	221	▌	245	⌡
6	♠	30	▲	54	6	78	N	102	f	126	~	150	û	174	«	198	╞	222	▐	246	÷
7		31	▼	55	7	79	O	103	g	127	⌂	151	ù	175	»	199	╟	223	▀	247	≈
8		32		56	8	80	P	104	h	128	Ç	152	ÿ	176	░	200	╚	224	α	248	°
9		33	!	57	9	81	Q	105	i	129	ü	153	Ö	177	▒	201	╔	225	ß	249	∙
10		34	"	58	:	82	R	106	j	130	é	154	Ü	178	▓	202	╩	226	Γ	250	·
11		35	#	59	;	83	S	107	k	131	â	155	¢	179	│	203	╦	227	π	251	√
12		36	$	60	<	84	T	108	l	132	ä	156	£	180	┤	204	╠	228	Σ	252	ⁿ
13		37	%	61	=	85	U	109	m	133	à	157	¥	181	╡	205	═	229	σ	253	²
14		38	&	62	>	86	V	110	n	134	å	158	₧	182	╢	206	╬	230	µ	254	■
15		39	'	63	?	87	W	111	o	135	ç	159	ƒ	183	╖	207	╧	231	τ	255	
16	►	40	(	64	@	88	X	112	p	136	ê	160	á	184	╕	208	╨	232	Φ		
17		41	)	65	A	89	Y	113	q	137	ë	161	í	185	╣	209	╤	233	Θ		
18	↕	42	*	66	B	90	Z	114	r	138	è	162	ó	186	║	210	╥	234	Ω		
19	‼	43	+	67	C	91	[	115	s	139	ï	163	ú	187	╗	211	╙	235	δ		
20	¶	44	,	68	D	92	\	116	t	140	î	164	ñ	188	╝	212	╘	236	∞		
21	§	45	-	69	E	93	]	117	u	141	ì	165	Ñ	189	╜	213	╒	237	φ		
22	▬	46	.	70	F	94	^	118	v	142	Ä	166	ª	190	╛	214	╓	238	ε		
23	↨	47	/	71	G	95	_	119	w	143	Å	167	º	191	┐	215	╫	239	∩		
24	↑	48	0	72	H	96	`	120	x	144	É	168	¿	192	└	216	╪	240	≡		

Figura 3.7: Tabela ASCII. Fonte: https://br.pinterest.com/pin/715931672003073964

Tanto na codificação como na decodificação, os dados do arquivo terão que ser lidos um-a-um, para que se possa comparar ao arquivo de dicionário já existente. Este pseudo-código dará origem a uma tabela que contém o dicionário.

Para a decodificação utiliza-se o processo inverso. Com a sequência de códigos que formam o arquivo codificado é feita uma busca na tabela, que é elaborada em paralelo com o processo de decodificação, com as sequências de símbolos que originaram estes códigos. É fácil verificar que a tabela pode ser montada como na codificação, bastando que esteja com as 256 palavras-código correspondentes às sequências unitárias (tabela ASCII). Nesse contexto, são necessários alguns pré-requisitos, sendo o mais relevante obtenção de uma estrutura de banco de dados adequada para a tabela de sequências, permitindo-se a busca e inserção eficientemente de uma sequência. Isto se deve ao fato de a tabela ser consultada e alterada para cada símbolo, tanto na compressão como na descompressão. No caso da compressão, a chave de busca na tabela é uma sequência de símbolos e a informação procurada é o seu código associado.

Por fim, a implementação do algoritmo de codificação/decodificação LZW, a definição do tamanho em bits das palavras-código é um fator crucial, por sua influência direta no tamanho das tabelas. Este tamanho das tabelas é uma função exponencial do tamanho das palavra-códigos, sendo que para cada bit adicional no comprimento destas, as tabelas dobram de tamanho. Logo, quanto maior for o tamanho das palavras-código, maior será a taxa de compressão. Este tamanho é limitado pela quantidade de memória disponível para construção da tabela. Uma limitação que reduz a portabilidade do algoritmo LZW consiste em que, quando o número de bits das palavras código cresce, tem-se que computadores com limitações de memória podem não conseguir decodificar arquivos codificados por um outro computador que possuía mais memória disponível para construção da tabela. Essas questões aqui abordadas são ilustradas nas Figuras 3.8 (codificador), 3.9 (tabela de encode) e 3.10 (decodificador).

```
Inicializar tabela com símbolos simples
P = primeira entrada de símbolo

ENQUANTO não receber última entrada
C = próximo entrada de símbolo
    SE (P + C) existir na tabela de símbolos
        P = P + C
    SENÃO
        Saída (P)
        Adicionar (P + C) ao dicionário
    FIM SE
FIM ENQUANTO
Saída(P)

```

Figura 3.8: Pseudo-código codificação LZW. Extraído de Salomon (2002).

A Tabela 3.1 apresenta um comparativo entre as diversas técnicas de compressão de dados, tipos de dados aplicáveis, nível de compressão, esforço computacional e limitação correspondente à entropia extraída dos trabalhos relacionados.

S	c	Saída (Ficheiro comprimido)	Código	Sequência
Símbolos dos dados de entrada (S_i)	Próximo símbolo dos dados de entrada (c_i)	Código de S	Código correspondente à sequência	Símbolo ou conjunto de símbolos que formam o dicionário

Código comprimido (Saída) — **Dicionário** (Código, Sequência)

Figura 3.9: Tabela LZW adaptado.

```
Inicializar tabela com símbolos simples
O = primeira entrada de símbolo
Saída(tradução(O))

ENQUANTO não receber última entrada
    N = próximo entrada de símbolo

    SE N não existir na tabela de símbolos
        S = tradução(O)
        S = S + N
    SENÃO
        S =tradução(N)
    FIM SE
    Saída(S)
    C = primeiro símbolo de S
    O + C adcionado a tabela de símbolos
    O = N
FIM ENQUANTO
```

Figura 3.10: Pseudo-código decodificação LZW adaptado.

Tabela 3.1: Comparativo entre trabalhos relacionados.

Método	Tipo de dado	Compressão	Esforço	Entropia

continua na próxima página...

Tabela 3.1 – continuando da página anterior.

Método	Tipo de dado	Compressão	Esforço	Sem Perdas
Huffman de variação mínima	Todos	Média	Baixo	Sim
Huffman limitado por comprimento	Alta	Alta	Baixo	Sim
Código Huffman não binário	Todos	Baixa	Baixo	Sim
Huffman adaptativo	Todos	Alta	Médio	Sim
Golomb	Imagem	Baixa	Alto	Não
Rice	Imagem	Baixa	Alto	Não
Tunstall	Todos	Baix a	Alto	Sim
LZ77	Todos	Baixa	Alto	Sim
LZW	Todos	Alta	Médio	Sim
LZ78	Todos	Alta	Médio	Sim
Burrows–Wheeler	Texto	Alta	Alto	Sim
Imagem Embutida	Imagem	Média	Alto	Não
Bits variável escalável	Áudio	Alta	Alto	Não
MPEG	Vídeo	Alta	Alto	Não

4

Modelo de Minimização de Entropia - MME

A busca por uma técnica de compressão de dados determinística e independente das limitações impostas pela entropia da informação representa um grande desafio. A principal motivação para esta busca é a existências de séries numéricas como a famosa Série de Ramanujan, definida na Equação 4.1 Ramanujan et al. (2013). Estas séries representam grupos de dados com distribuição estatística uniforme ou próximas a isso, como por exemplo os dígitos de π. Isso naturalmente sugere questionamentos quanto à síntese de séries numéricas que representem estes conjuntos de dados, ou sua representação como coordenadas num espaço N dimensional, e principalmente se tais abordagens possibilitam a compressão de dados.

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{99^2} \sum_{k=0}^{\infty} \frac{(4k)!}{k!^4} \frac{26390k + 1103}{396^{4k}} \tag{4.1}$$

Através da aplicação da série de Ramanujan, o número π que é transcendental e com infinitos dígitos, pode ser obtido de forma iterativa em grupos de oito dígitos, conforme discutido por Shidlovskii (2011). Porém esta constante universal apresenta entropia próxima a 0,5 quando analisada através das equações de Shannon (Shannon (1948)). Neste contexto, observa-se o mesmo conjunto de dados com entropia alta se tratado estatisticamente, e entropia zero caso iterativamente seja aplicada a Equação 4.1. Com base nisso, neste trabalho conjecturou-se a existência de modelos representativos geométrica ou numéricos de grupos de dados estatisticamente dispersos ou uniformes, o que fez surgir a seguinte questão científica: *Qual modelo permitirá a síntese de séries numéricas que representem sequência de dados ?*

4.1 Breve Histórico

Como decorrência de questões como essas, após um período significativo de pesquisas, a abordagem utilizada nesta pesquisa procura codificar um conjunto de dados aplicando transformações e convertendo-os em espaços diferentes. A pesquisa para alcançar o modelo que será apresentado a seguir se iniciou em 1992, com diversas abordagens estudadas e testadas em um ciclo de evolução natural de conversões de espaços que convergiu ao modelo atual (2022), intensificados nos últimos 4 anos. Nesse contexto, lista-se um breve histórico desta busca, onde a identificação da estrutura atual é decorrência das pesquisas e testes destes modelos.

1. Modelo de orbitais (1995), baseado nas equações de *Schrödinger*. Neste modelo, as sequências de dados sofria uma transformação em sua representação para orbitais e seus valores eram encontrados pelas equações descritivas destes orbitais.

2. Representação de dados por atratores estranhos (2001), onde a representação dos dados sofria transformação através das equações diferenciais acopladas deste atrator.

3. Aplicação do fractal *Mandelbrot* à codificação de dados (2005), onde a compressão esperada seria decorrência da generalização da codificação fractal Conci and Aquino (2005).

4. Redes Neurais evolutivas (2008), onde a compressão de dados seria obtida a partir da redução de dimensionalidade obtida através desta rede neural.

5. Representação de dados no R2 (2015), onde a compressão de dados seria obtida a partir do ponto médio entre coordenadas no plano. Este modelo é base para o modelo atual.

6. Séries Ramanujan/Sato Ramanujan et al. (2013), como representações de conjuntos de dados, onde se buscou generalizar a representação de números pseudo-aleatórios por séries numéricas. Através do estudo da geometria envolvida nestas séries em conjunto com a representação de dados no R2, o MME é decorrência direta do aprendizado especialmente deste modelo.

Em suma, a limitação encontrada nessas tentativas foi que os modelos apresentam convergência não garantida (nem sempre a decodificação produzia o resultado esperado). Em cada caso, geralmente o esforço computacional necessário à conversão do conjunto de dados é proibitivo em escalas comerciais e não simétrico (pelo menos ao ponto do que foi possível alcançar).

4.2 Introdução ao MME

O MME é um codificador determinístico e iterativo operando em um espaço geométrico baseado em círculos verticais acoplados aos seus complementares horizontais integrados a uma esfera,

(Figura 4.1). Nele, as coordenadas determinam o conjunto de dados codificados através de iterações sucessivas, onde cada símbolo é convertido de acordo com o alfabeto utilizado, em uma posição normalizada, permitindo sua representação como um ponto no espaço. Nesta abordagem a incerteza tende a zero, uma vez que a $p_i = 1$, conforme Equação 4.2 (discutida no Capítulo **??**).

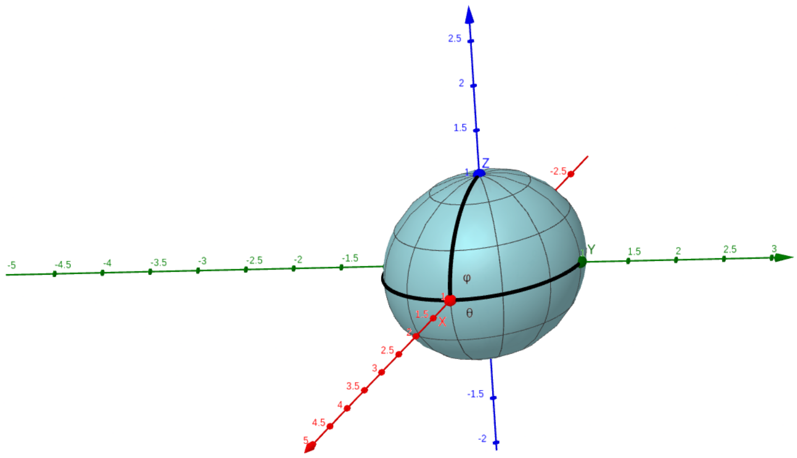


Figura 4.1: Mapeamento de eixos em esfera.

$$H(p_1,...,p_n) = -\sum_{i=1}^{n} p_i \log(p_i) = 0 \tag{4.2}$$

A minimização da entropia da informação a zero possibilita a utilização desta abordagem de codificação em compressão de dados sem perdas a qualquer conjunto de dados que atenda aos seguintes requisitos:

1. Os símbolos utilizados no conjunto de dados formam um conjunto finito.
2. A quantidade de símbolos utilizada no conjunto de dados é menor que a quantidade de símbolos representáveis pela precisão numérica utilizada.
3. As distâncias entre os pontos normalizados a partir dos símbolos devem ser iguais.

Conforme observa-se na Figura 4.2, padronizou-se o espaço MME como uma esfera de raio 1,0, onde infinitos círculos verticais e horizontais representam a informação codificada. A informação é representada verticalmente e sua transferência ao próximo estado é representada através da projeção do ponto médio ao plano perpendicular à esfera. O ponto médio entre dois pontos previamente normalizados, interior à esfera, concentra a informação referente aos pontos codificados e sua projeção transfere esta informação ao próximo ciclo de codificação.

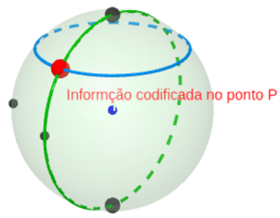


Figura 4.2: Ponto P de interseção entre círculos verticais e horizontais.

A eliminação de redundância de informações em MME é obtida através da estimativa das coordenadas que representam toda a série de dados no estado atual, sendo necessária a cada iteração uma nova estimativa de coordenadas, sempre cumulativa em relação à série. Na Figura 4.3, ilustra-se o fluxo de codificação.

Assim, as operações de codificação buscam um lugar geométrico onde as coordenadas no espaço representam a série de dados e possibilitam sua decodificação. Nesta abordagem, o dicionário resultante é o conjunto de coordenadas no espaço que representam a última iteração. O processo de decodificação é a estimação da coordenadas espaciais a partir dos pontos conhecidos que, desta forma, disponibiliza os dados acumulados pela relação posição/iteração. O MME possibilita a compressão de dados máxima possível para um precisão numérica desejada, independentemente do tamanho da série de dados a receber a compressão. Na Figura 4.4, ilustra-se o fluxo de execução de decodificação.

4.3 Representação de um símbolo no plano

Seja uma circunferência C, orientada positivamente no sentido anti-horário, a partir do ponto (0, 0), e $C = (x,y) \in \mathbb{R}|x2+y2 = 1$ dividida em quatro quadrantes.

Sendo assim, pode-se associar todo número real, entre 0 e 2π, a um ponto do círculo trigonométrico com um ângulo φ. Seja um ponto B um ponto no círculo trigonométrico com ângulo φ, sua posição (x,y) será definida pelo tupla (sen(B) e cos(B)).

Seja M um conjunto qualquer, com M $\neq \emptyset$ e seja d : M x M $\rightarrow \mathbb{R}$ uma função. Sendo d(x, y) a imagem de um par (x, y) $\in$ M x M, através da função d. Se d satisfaz as propriedades a seguir, então d é chamada uma métrica sobre M.

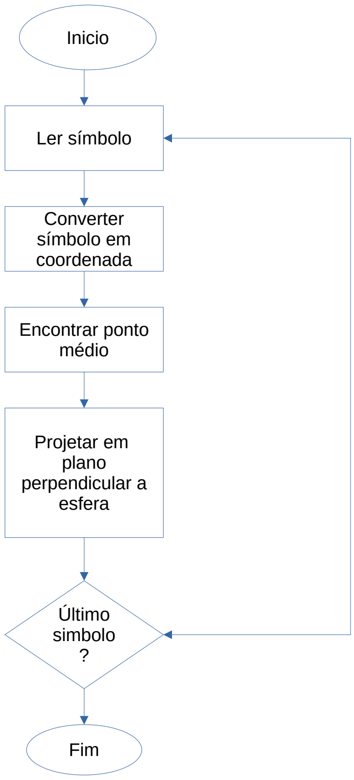


Figura 4.3: Fluxo de codificação.

1. d(x, y) = 0
2. d(x, y) > 0 se x $\neq$ 0
3. d(x, y) = d(y,x)
4. d(x, z) $\leq$ d(x,y) + d(y,z), para todo x,y,z $\in$ M

Nestas condições, cada imagem d(x, y) recebe o nome de distância de x a y e uma par (M,d), onde d é uma métrica sobre M e denominado *espaço métrico*. Considerando o conjunto S de todos os símbolos utilizados na representação de um conjunto de dados, é aplicada uma métrica M, onde S é um conjunto qualquer, sendo φS_n a imagem do ângulo $\in$ S x S, através da função φ. Se φ satisfaz aos seguintes requisitos:

1. $S \neq \emptyset$
2. $\varphi S_n > 0$
3. $\varphi S_n < \frac{\pi}{2}$

A circunferência se apresenta como a figura geométrica adequada à codificação de sequências de dados devido à sua simetria perfeita em cada ponto, permitindo-se a obtenção de pontos em quadrantes inversos.

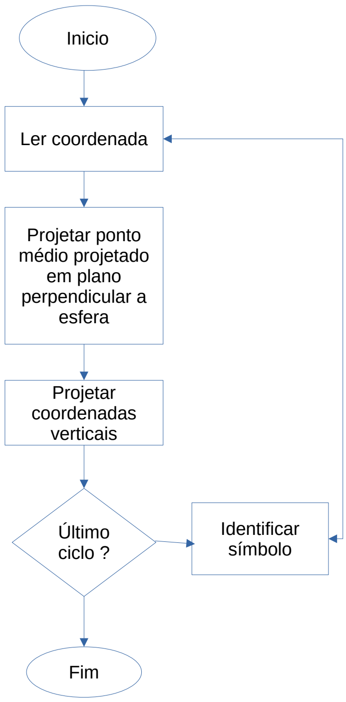


Figura 4.4: Fluxo de decodificação.

A conversão do conjunto de símbolos S ocorre naturalmente pela divisão da área dos seus quadrantes pela quantidade de símbolos constantes no conjunto S. A representação geométrica dos símbolos através de coordenadas no espaço é obtida, através da divisão do espaço métrico pelo quantidade de símbolos do conjunto de dados a ser codificado.

Sendo S o conjunto de símbolos que representam uma série de dados, em MME o valor de cada ângulo φ é determinado pela divisão da área disponível em $\frac{\pi}{2}$ pela quantidade de símbolos utilizados em S, multiplicado pela posição do símbolo utilizado em S_n e suas coordenadas pelos cossenos e senos respectivos, conforme Equações 4.3 e 4.4.

$$\varphi = \frac{\pi}{S} S_n \tag{4.3}$$

$$\varphi(x, y) = \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \tag{4.4}$$

4.3.1 Conversão de Alfabeto em coordenadas no espaço

A posição em θ é determinada pelo ângulo em que o ponto codificado se posicionará longitudinalmente, conforme ilustra-se na Figura 4.5, sendo sempre deslocado a partir da projeções deste ponto sobre os planos paralelos à esfera.

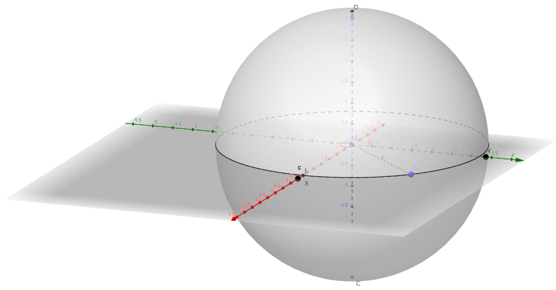


Figura 4.5: Deslocamento do ponto codificado.

Neste trabalho, assume-se o ângulo zero na primeira iteração, sendo os ângulos posteriores resultantes das iterações subsequentes ao processo de codificação. Desta forma, as coordenadas para o ponto serão dados pela Equação 4.5.

$$P_(x,y,z) = \begin{pmatrix} \cos\varphi \sin\theta \\ \sin\varphi \sin\theta \\ \cos\varphi \end{pmatrix} \tag{4.5}$$

Assim, um símbolo codificado no espaço será representado por um ponto na superfície de uma esfera, conforme ilustra-se na Figura 4.6.

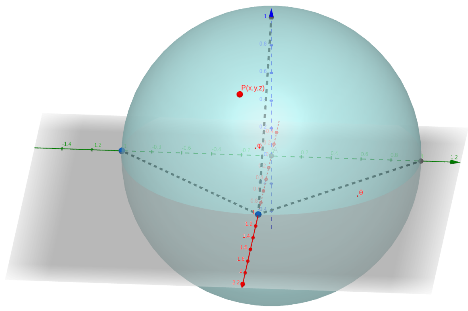


Figura 4.6: Caracterização de símbolo de alfabeto no espaço.

4.4 Codificador

O processo de codificação consiste em condensar a informação das coordenadas de dois símbolos em um ponto médio das coordenadas φ. Ao aplicar φ ao plano, obtém-se a posição do ponto codificado no espaço.

$$P_m(x,y,z) = \begin{pmatrix} \frac{\cos\varphi_1 \sin\theta_1 + \cos\varphi_2 \sin\theta_2}{2} \\ \frac{\sin\varphi_1 \sin\theta_1 + \sin\varphi_2 \sin\theta_2}{2} \\ \frac{\cos\varphi_1 + \cos\varphi_2}{2} \end{pmatrix} \tag{4.6}$$

O ponto $P_m(x,y,z)$, encontrado terá coordenadas interiores à esfera, conforme Figura 4.7. Para normalização dos valores é necessária a projeção ortogonal do ponto P_m ao plano perpendicular à esfera, através de equação paramétrica. O pontos projetados com ângulo de 90 graus formam assim um quadrilátero regular inscrito na esfera. Seus vetores são definidos pelas Equações 4.7 e 4.8.

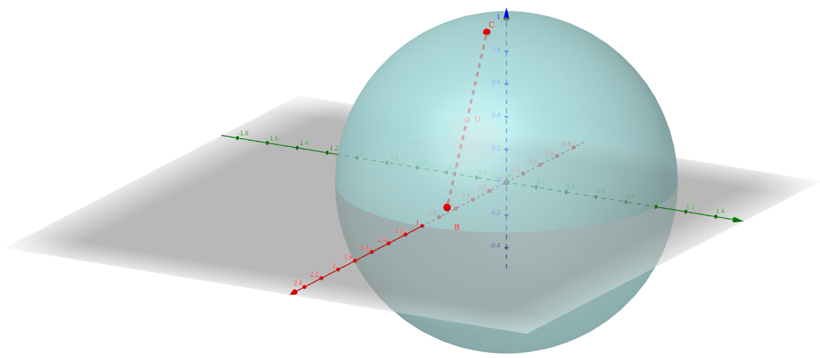


Figura 4.7: Ponto D interior à esfera.

$$v_(x,y,z) = \begin{pmatrix} \sin\theta_m \\ \cos\theta_m \\ \cos\theta_m + \sin\theta_m \end{pmatrix} \tag{4.7}$$

$$v1_(x,y,z) = \begin{pmatrix} P_m(x) + v(x) \\ P_m(y) + v(y) \\ P_m(z) + v(z) \end{pmatrix} \tag{4.8}$$

Assim, são obtidos os parâmetros da equação quadrática através das Equações 4.9, 4.10 e 4.11.

$$a = (v1(x) - P_m(x))^2 + (v1(y) - P_m(y))^2 + (v1(z) - P_m(z))^2 \tag{4.9}$$

$$b=((v1(x)-v(x))^2\cdot(0-P_m(x)))+((v1(y)-v(y))^2\cdot(0-P_m(y)))+(v1(z)-v(z))^2\cdot(0-P_m(z))) \tag{4.10}$$

$$c=(1-((0-P_m(x))^2+(0-P_m(y))^2+(0-P_m(z))^2))^2 \tag{4.11}$$

As raízes da equação quadrática correspondem aos últimos parâmetros necessários à projeção dos ponto médio aos planos paralelos à esfera, conforme Equações 4.12 e 4.13.

$$T_1=-b+\sqrt{\frac{b^2-4ac}{2a}} \tag{4.12}$$

$$T_2=-b-\sqrt{\frac{b^2-4ac}{2a}} \tag{4.13}$$

Os pontos projetados F e G são obtidos através das Equações 4.14 e 4.15 e sua projeção ortogonal do ponto médio em planos paralelos à esfera.

Na Figura **??** ilustra-se a

$$F_(x,y,z)=\begin{pmatrix} P_m(x)+(v1(x)-P_m(x))\cdot T_1 \\ P_m(y)+(v1(y)-P_m(y))\cdot T_1 \\ P_m(z)+(v1(z)-P_m(z))\cdot T_1 \end{pmatrix} \tag{4.14}$$

$$G_(x,y,z)=\begin{pmatrix} P_m(x)+(v1(x)-P_m(x))\cdot T_2 \\ P_m(y)+(v1(y)-P_m(y))\cdot T_2 \\ P_m(z)+(v1(z)-P_m(z))\cdot T_2 \end{pmatrix} \tag{4.15}$$

As operações de codificação no MME buscam um lugar geométrico onde as coordenadas no espaço representam o conjunto de dados e possibilitam sua decodificação com base nos pontos F e G, conforme ilustra-se na Figura 4.8.

Os pontos F e G obtidos apresentam ângulo de 90 graus a partir do ponto médio P_m, porém o inverso não é verdade, sendo necessária a identificação do ângulo inverso de F e G a P_m. Este processo é cumulativo, uma vez que para cada iteração a informação destes ângulos é transferida à próxima coordenada. A convergência dos ângulos de retorno é feita através do

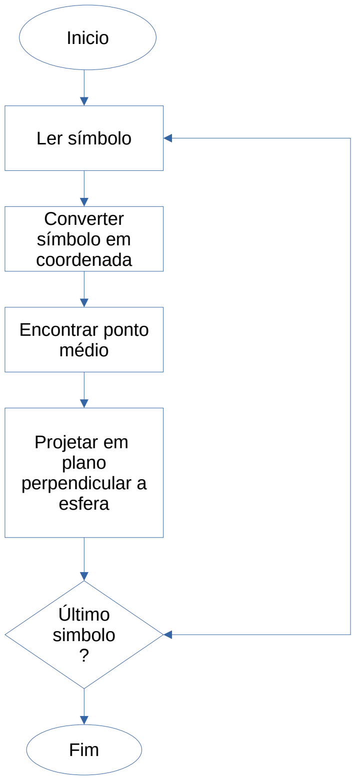


Figura 4.8: Fluxo de codificação.

Algoritmo de relação inteira Borwein et al. (2004) e definido pela Equação 4.16, onde A e B são os número inteiros esperados Bailey and Borwein (2009). O o parâmetro P_θ acumula a informação de recuperação do ângulo de P_m em relação a F ou G nas Equações 4.17 e 4.18.

$$\theta_F + A(2\pi) = \theta_G + B(2\pi) \tag{4.16}$$

$$P_\theta = \theta_F + A(2\pi) \tag{4.17}$$

$$P_\theta = \theta_G + B(2\pi) \tag{4.18}$$

4.5 Decodificador

No processo de decodificação, as coordenadas devem ser encontradas a partir dos pontos projetados e, através dos pontos F ou G, encontra-se o ponto médio. Deve-se observar que o sentido de deslocamento angular de θ é inverso ao processo de codificação. Caso tenha sido utilizado o ponto G para os ciclos seguintes de codificação, o processo inverso deve iniciar neste ponto e buscar o ponto médio vinculado a ele. Na Equação 4.19 é aplicada a função de recuperação no ponto G para obtenção das coordenadas x, y e z. Uma vez encontrado o ponto médio P_m, as coordenadas de x e y serão utilizadas para identificação das coordenadas do símbolo codificado e do ponto de conexão com o próximo ciclo de decodificação. As Equações 4.20, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28 e 4.29 são utilizadas para identificação da posição original do símbolos codificados e sua posterior conversão em seu valor. As operações de decodificação limitam-se à quantidade de ciclos executados no processo de codificação, conforme ilustra-se no Fluxo da Figura 4.9.

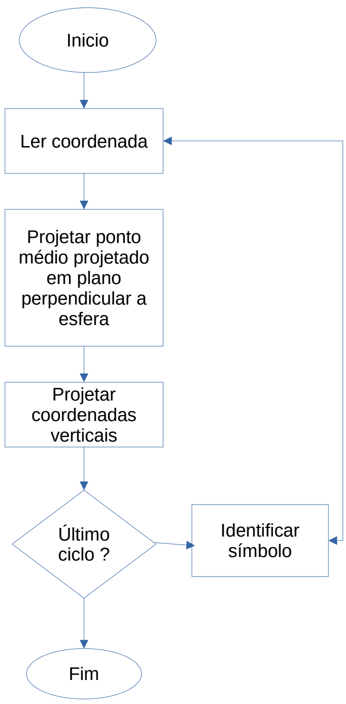


Figura 4.9: Fluxo de decodificação.

$$P_m(x,y,z) = \begin{pmatrix} P_\theta : G_x \rightarrow x \\ P_\theta : G_y \rightarrow y \\ G_z \end{pmatrix} \tag{4.19}$$

$$d_(P_m, C_1) = \sqrt{P_m(x)^2 + P_m(y)^2} \tag{4.20}$$

$$r1 = \sqrt{1 - d_(P_m, C_1)^2} \tag{4.21}$$

$$a = ((1 - d_(P_m, C_1)^2) + d^2)/2d \tag{4.22}$$

$$h = \sqrt{1 - a^2} \tag{4.23}$$

$$X = a\frac{P_m(x)}{d} \tag{4.24}$$

$$Y = a\frac{P_m(y)}{d} \tag{4.25}$$

$$P(x_1) = X + h\frac{P_m(y)}{d} \tag{4.26}$$

$$P_(y_1) = Y - h\frac{P_m(x)}{d} \tag{4.27}$$

$$P_(x_2) = X - h\frac{P_m(y)}{d} \tag{4.28}$$

$$P_{(y_2)} = Y + h\frac{P_m(x)}{d} \tag{4.29}$$

Para a identificação dos Parâmetros A e B necessários às Equações 4.17 e 4.18, faz-se necessário a utilização de algoritmo para detecção de relações inteiras, que tem se tornado expressivo em matemática experimental. Esta técnica utiliza computação para explorar questões matemáticas e físicas. A detecção de relação entre inteiros, apresentada em (Bailey and Borwein (2009)), encontra uma relação em um número limitado de iterações. Após encerrada a busca os parâmetros A e B nas Equações 4.17 e 4.18, fornecerão a velocidade angular necessária a recuperação das coordenadas de centros médios anteriormente calculadas. O processo deve se repetir até o ultimo ciclo de processamento para recuperação do conjunto de dados original.

5

Resultados

Neste capítulo são descritos os experimentos de codificação e análise de entropia e conjunto de dados resultante. Além disso, são apresentadas as metodologias adotadas para obtenção dos valores finais para cada uma das métricas de interesse durante a execução dos experimentos. Através dos valores resultantes de entropia $H(i)$, conjunto de dados resultante e esforço computacional, foi possível realizar comparações quanto ao desempenho de cada um dos métodos analisados e compará-los com o modelo MME.

5.1 Metodologia

Os experimentos foram elaborados com o objetivo de verificar o comportamento do Modelo MME, quanto à sua independência das limitações relativas à entropia da informação, bem como o resultado dos processos iterativos, sendo esperada uma saída de tamanho fixo e independente do tamanho do conjunto de amostras de entrada. O Algoritmo LZW((Lempel–Ziv–Welch) foi utilizado como parâmetro de comparação por ser referência entre algoritmos de compressão de dados, apresentar implementação simples e esforço computacional considerado pequeno.

Para verificação da abrangência do modelo MME, foram realizados experimentos, onde se buscou verificar a redução da entropia a zero através da aplicação do modelo MME e sua saída resultante, sendo esperado um conjunto de coordenadas no espaço que permita a reconstrução dos dados de entrada sem perdas e de forma iterativa. Nestes experimentos foram os comparados os conjunto de dados resultantes e o esforço computacional após a aplicação do Modelo MME e do Algoritmo LZW((Lempel–Ziv–Welch). Para atendimento aos objetivos já mencionados, bucou-se três conjuntos de dados que permitissem esta verificação, sendo:

- Conjunto de 1.024 amostras uniformemente distribuídas - Este conjunto representa o limite de variância e incerteza na distribuição dos dados, e desta forma, algoritmos limitados à entropia da informação tenderão a apresentar baixa capacidade de compressão nestes cenários. Buscou-se desta forma verificar a independência do Modelo MME das limitações da entropia da informação.

- Imagem de formato PNG (Portable Network Graphic) - Este formato de imagem está fortemente disseminado na Internet e apresenta baixa previsibilidade de compressão de dados, uma vez que sua distribuição dos dados pode variar de uniforme a aglomeração em grupos. Buscou-se desta forma verificar a estabilidade do Modelo MME em fornecer uma saída de formato único e independente do tipo do conjunto de dados de entrada, bem como sua distribuição.

- Conjunto de 1.00 amostras iniciais dos dígitos do número π. Este conjunto representa um número transcendental (Winitzki (2003)). Buscou-se desta forma, verificar a possibilidade de reconstrução de um número através da saída fixa do Modelo MME, o que permitirá em futuros trabalhos a identificação de séries numéricas que descrevam números transcedentais.

5.2 Amostras uniformemente distribuídas

Neste experimento, foi utilizada um conjunto de 1.024 amostras de dados pseudo-aleatórios, uniformemente distribuídas com um conjunto de 10 símbolos passíveis de codificação. Apresentando uma entropia de informação $H(i) = 0,4685233730$, conforme Tabela 5.1.

Tabela 5.1: Primeiro experimento. Símbolos utilizados e entropia calculada.

Experimento com 1024 amostras uniformemente distribuídas.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
0	93	0,0908203125	0,43922020026
1	107	0,1044921875	0,4830758560
2	105	0,0908203125	0,4769929439
3	95	0,0927734375	0,4456600592
4	106	0,1035156250	0,4800418130
5	111	0,1083984375	0,4950661818
6	113	0,1103515625	0,5009754669
7	107	0,1044921875	0,4830758560
8	88	0,0859375000	0,4227623438
9	99	0,0966796875	0,4583812069
Ocorrências	1.024		0,4685233730

Para efeito comparativo, aplicou-se a codificação LZW ao mesmo conjunto de dados. O código-fonte do LZW utilizado teste trabalho está disponível em Ali (2014). A característica do algoritmo LZW é a eliminação de redundâncias através da utilização de dicionários de sequências e seu posicionamento do arquivo digital. Para isso, todos os dados são convertido para a classe de 256 símbolos de representação, conforme tabela ASCII utilizada como padrão na codificação de símbolos em arquivos digitais.

As sequências de símbolos encontradas e suas respectivas frequências e entropia são ilustradas na Tabela 5.2. Como resultado da aplicação do algoritmo LZW ao conjunto de dados analisado, constatou-se uma redução na entropia devido ao aumento da quantidade de símbolos representada de 10 para 256. Neste contexto, a entropia identificada foi $0,14400849016374$. Também foi observada uma redução na quantidade de símbolos utilizados para 619, caracterizando uma redução de 405 símbolos na representação da informação contida nestes dados.

Tabela 5.2: Entropia após aplicação codificação LZW.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
0	0	0	0
1	0	0	0
2	1	0,00161550888529887	0,173107022111895
3	4	0,00646203554119548	0,562960852573012
4	2	0,00323101777059774	0,313866153125025
5	3	0,00484652665589661	0,442392234950034
6	6	0,00969305331179321	0,787513413034796
7	2	0,00323101777059774	0,313866153125025
8	0	0	0
9	2	0,00323101777059774	0,313866153125025
10	2	0,00323101777059774	0,313866153125025
11	1	0,00161550888529887	0,173107022111895
12	4	0,00646203554119548	0,562960852573012
13	9	0,0145395799676898	1,09570657683356
14	4	0,00646203554119548	0,562960852573012
15	0	0	0
16	5	0,00807754442649435	0,677602590983891
17	3	0,00484652665589661	0,442392234950034
18	8	0,012924071082391	0,996074632542585

continua na próxima página...

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
19	1	0,00161550888529887	0,173107022111895
20	4	0,00646203554119548	0,562960852573012
21	3	0,00484652665589661	0,442392234950034
22	2	0,00323101777059774	0,313866153125025
23	3	0,00484652665589661	0,442392234950034
24	3	0,00484652665589661	0,442392234950034
25	7	0,0113085621970921	0,893483549779059
26	3	0,00484652665589661	0,442392234950034
27	2	0,00323101777059774	0,313866153125025
28	7	0,0113085621970921	0,893483549779059
29	3	0,00484652665589661	0,442392234950034
30	0	0	0
31	1	0,00161550888529887	0,173107022111895
32	1	0,00161550888529887	0,173107022111895
33	6	0,00969305331179321	0,787513413034796
34	3	0,00484652665589661	0,442392234950034
35	3	0,00484652665589661	0,442392234950034
36	5	0,00807754442649435	0,677602590983891
37	6	0,00969305331179321	0,787513413034796
38	1	0,00161550888529887	0,173107022111895
39	3	0,00484652665589661	0,442392234950034
40	2	0,00323101777059774	0,313866153125025
41	3	0,00484652665589661	0,442392234950034
42	2	0,00323101777059774	0,313866153125025
43	1	0,00161550888529887	0,173107022111895
44	1	0,00161550888529887	0,173107022111895
45	1	0,00161550888529887	0,173107022111895
46	0	0	0
47	0	0	0
48	3	0,00484652665589661	0,442392234950034
49	9	0,0145395799676898	1,09570657683356
50	2	0,00323101777059774	0,313866153125025
51	2	0,00323101777059774	0,313866153125024
52	4	0,00646203554119548	0,562960852573013
continua na próxima página...			

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
53	7	0,0113085621970921	0,893483549779058
54	2	0,00323101777059774	0,313866153125024
55	5	0,00807754442649435	0,677602590983891
56	1	0,00161550888529887	0,173107022111895
57	3	0,00484652665589661	0,442392234950034
58	1	0,00161550888529887	0,173107022111895
59	1	0,00161550888529887	0,173107022111895
60	2	0,00323101777059774	0,313866153125024
61	2	0,00323101777059774	0,313866153125024
62	0	0	0
63	2	0,00323101777059774	0,313866153125024
64	1	0,00161550888529887	0,173107022111895
65	1	0,00161550888529887	0,173107022111895
66	3	0,00484652665589661	0,442392234950034
67	4	0,00646203554119548	0,562960852573013
68	3	0,00484652665589661	0,442392234950034
69	3	0,00484652665589661	0,442392234950034
70	9	0,0145395799676898	1,09570657683356
71	1	0,00161550888529887	0,173107022111895
72	3	0,00484652665589661	0,442392234950034
73	3	0,00484652665589661	0,442392234950034
74	4	0,00646203554119548	0,562960852573013
75	2	0,00323101777059774	0,313866153125024
76	5	0,00807754442649435	0,677602590983891
77	5	0,00807754442649435	0,677602590983891
78	3	0,00484652665589661	0,442392234950034
79	0	0	0
80	6	0,00969305331179321	0,787513413034796
81	6	0,00969305331179321	0,787513413034796
82	1	0,00161550888529887	0,173107022111895
83	3	0,00484652665589661	0,442392234950034
84	5	0,00807754442649435	0,677602590983891
85	8	0,012924071082391	0,996074632542585
86	5	0,00807754442649435	0,677602590983891

continua na próxima página...

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
87	4	0,00646203554119548	0,562960852573013
88	5	0,00807754442649435	0,677602590983891
89	3	0,00484652665589661	0,442392234950034
90	4	0,00646203554119548	0,562960852573013
91	3	0,00484652665589661	0,442392234950034
92	2	0,00323101777059774	0,313866153125024
93	2	0,00323101777059774	0,313866153125024
94	1	0,00161550888529887	0,173107022111895
95	2	0,00323101777059774	0,313866153125024
96	5	0,00807754442649435	0,677602590983891
97	5	0,00807754442649435	0,677602590983891
98	2	0,00323101777059774	0,313866153125024
99	3	0,00484652665589661	0,442392234950034
100	7	0,0113085621970921	0,893483549779058
101	3	0,00484652665589661	0,442392234950034
102	3	0,00484652665589661	0,442392234950034
103	7	0,0113085621970921	0,893483549779058
104	5	0,00807754442649435	0,677602590983891
105	2	0,00323101777059774	0,313866153125024
106	4	0,00646203554119548	0,562960852573013
107	2	0,00323101777059774	0,313866153125024
108	2	0,00323101777059774	0,313866153125024
109	5	0,00807754442649435	0,677602590983891
110	2	0,00323101777059774	0,313866153125024
111	2	0,00323101777059774	0,313866153125024
112	2	0,00323101777059774	0,313866153125024
113	5	0,00807754442649435	0,677602590983891
114	2	0,00323101777059774	0,313866153125024
115	2	0,00323101777059774	0,313866153125024
116	1	0,00161550888529887	0,173107022111895
117	4	0,00646203554119548	0,562960852573013
118	2	0,00323101777059774	0,313866153125024
119	1	0,00161550888529887	0,173107022111895
120	1	0,00161550888529887	0,173107022111895

continua na próxima página...

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
121	3	0,00484652665589661	0,442392234950034
122	0	0	0
123	2	0,00323101777059774	0,313866153125024
124	0	0	0
125	3	0,00484652665589661	0,442392234950034
126	2	0,00323101777059774	0,313866153125024
127	0	0	0
128	2	0,00323101777059774	0,313866153125024
129	3	0,00484652665589661	0,442392234950034
130	4	0,00646203554119548	0,562960852573013
131	2	0,00323101777059774	0,313866153125024
132	4	0,00646203554119548	0,562960852573013
133	2	0,00323101777059774	0,313866153125024
134	3	0,00484652665589661	0,442392234950034
135	1	0,00161550888529887	0,173107022111895
136	5	0,00807754442649435	0,677602590983891
137	2	0,00323101777059774	0,313866153125024
138	2	0,00323101777059774	0,313866153125024
139	0	0	0
140	4	0,00646203554119548	0,562960852573013
141	4	0,00646203554119548	0,562960852573013
142	3	0,00484652665589661	0,442392234950034
143	0	0	0
144	6	0,00969305331179321	0,787513413034796
145	5	0,00807754442649435	0,677602590983891
146	4	0,00646203554119548	0,562960852573013
147	5	0,00807754442649435	0,677602590983891
148	3	0,00484652665589661	0,442392234950034
149	4	0,00646203554119548	0,562960852573013
150	2	0,00323101777059774	0,313866153125024
151	1	0,00161550888529887	0,173107022111895
152	8	0,012924071082391	0,996074632542585
153	0	0	0
154	0	0	0

continua na próxima página...

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
155	1	0,00161550888529887	0,173107022111895
156	0	0	0
157	3	0,00484652665589661	0,442392234950034
158	0	0	0
159	1	0,00161550888529887	0,173107022111895
160	0	0	0
161	2	0,00323101777059774	0,313866153125024
162	1	0,00161550888529887	0,173107022111895
163	1	0,00161550888529887	0,173107022111895
164	3	0,00484652665589661	0,442392234950034
165	2	0,00323101777059774	0,313866153125024
166	2	0,00323101777059774	0,313866153125024
167	0	0	0
168	4	0,00646203554119548	0,562960852573013
169	3	0,00484652665589661	0,442392234950034
170	4	0,00646203554119548	0,562960852573013
171	1	0,00161550888529887	0,173107022111895
172	1	0,00161550888529887	0,173107022111895
173	3	0,00484652665589661	0,442392234950034
174	1	0,00161550888529887	0,173107022111895
175	2	0,00323101777059774	0,313866153125024
176	2	0,00323101777059774	0,313866153125024
177	1	0,00161550888529887	0,173107022111895
178	0	0	0
179	0	0	0
180	3	0,00484652665589661	0,442392234950034
181	3	0,00484652665589661	0,442392234950034
182	2	0,00323101777059774	0,313866153125024
183	0	0	0
184	0	0	0
185	2	0,00323101777059774	0,313866153125024
186	0	0	0
187	0	0	0
188	0	0	0

continua na próxima página...

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
189	2	0,00323101777059774	0,313866153125024
190	1	0,00161550888529887	0,173107022111895
191	0	0	0
192	3	0,00484652665589661	0,442392234950034
193	3	0,00484652665589661	0,442392234950034
194	2	0,00323101777059774	0,313866153125024
195	4	0,00646203554119548	0,562960852573013
196	6	0,00969305331179321	0,787513413034796
197	1	0,00161550888529887	0,173107022111895
198	3	0,00484652665589661	0,442392234950034
199	1	0,00161550888529887	0,173107022111895
200	4	0,00646203554119548	0,562960852573013
201	0	0	0
202	1	0,00161550888529887	0,173107022111895
203	0	0	0
204	1	0,00161550888529887	0,173107022111895
205	2	0,00323101777059774	0,313866153125024
206	0	0	0
207	0	0	0
208	4	0,00646203554119548	0,562960852573013
209	2	0,00323101777059774	0,313866153125024
210	1	0,00161550888529887	0,173107022111895
211	3	0,00484652665589661	0,442392234950034
212	4	0,00646203554119548	0,562960852573013
213	5	0,00807754442649435	0,677602590983891
214	1	0,00161550888529887	0,173107022111895
215	1	0,00161550888529887	0,173107022111895
216	4	0,00646203554119548	0,562960852573013
217	3	0,00484652665589661	0,442392234950034
218	3	0,00484652665589661	0,442392234950034
219	1	0,00161550888529887	0,173107022111895
220	3	0,00484652665589661	0,442392234950034
221	0	0	0
222	0	0	0

continua na próxima página...

Tabela 5.2 – continuando da página anterior.

Entropia após aplicação codificação LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
223	2	0,00323101777059774	0,313866153125024
224	0	0	0
225	3	0,00484652665589661	0,442392234950034
226	0	0	0
227	2	0,00323101777059774	0,313866153125024
228	3	0,00484652665589661	0,442392234950034
229	4	0,00646203554119548	0,562960852573013
230	3	0,00484652665589661	0,442392234950034
231	2	0,00323101777059774	0,313866153125024
232	2	0,00323101777059774	0,313866153125024
233	2	0,00323101777059774	0,313866153125024
234	2	0,00323101777059774	0,313866153125024
235	1	0,00161550888529887	0,173107022111895
236	0	0	0
237	2	0,00323101777059774	0,313866153125024
238	1	0,00161550888529887	0,173107022111895
239	1	0,00161550888529887	0,173107022111895
240	4	0,00646203554119548	0,562960852573013
241	0	0	0
242	0	0	0
243	1	0,00161550888529887	0,173107022111895
244	0	0	0
245	3	0,00484652665589661	0,442392234950034
246	0	0	0
247	0	0	0
248	2	0,00323101777059774	0,313866153125024
249	0	0	0
250	0	0	0
251	0	0	0
252	0	0	0
253	3	0,00484652665589661	0,442392234950034
254	1	0,00161550888529887	0,173107022111895

Foi aplicado ao mesmo conjunto de dados anteriormente codificado ao Modelo MME, obtendo-se as coordenadas no espaço (Figura 5.1 e Tabela 5.3) que representam a informação nele contida. Como procedimento padronização ao algoritmo LZW, utilizou-se a quantidade de 256 símbolos para a representação da informação.

```
Cicle 1021
Ponto C     ( -0.1575687867232671 , 0.2767416502948283 , 0.9479378336381835 )
Ponto D     ( -0.1682844414062585 , 0.2955617987685784 , 0.9403847988399568 )
Ponto medio ( -0.1629266140647628 , 0.2861517245317034 , 0.944161316239070 )
Ponto F     ( -0.1529606736495757 , 0.2918260461891209 , 0.9441613162390700 )
Ponto G     ( -0.1728925544799501 , 0.2804774028742858 , 0.9441613162390700 )
============================================================
************************************************************
Cicle 1022
Ponto C     ( -0.1728925544799501 , 0.2804774028742858 , 0.9441613162390700 )
Ponto D     ( -0.1694276335765134 , 0.2748563857108575 , 0.9464397731576093 )
Ponto medio ( -0.1711600940282318 , 0.2776668942925716 , 0.9453005446983395 )
Ponto F     ( -0.1681869745734192 , 0.2794995917293308 , 0.9453005446983395 )
Ponto G     ( -0.1741332134830471 , 0.2758341968558107 , 0.9453005446983395 )
============================================================
************************************************************
Cicle 1023
Ponto C     ( -0.1741332134830471 , 0.2758341968558107 , 0.9453005446983395 )
Ponto D     ( -0.1600067836873424 , 0.2534573490437329 , 0.9540215937754005 )
Ponto medio ( -0.1670699985851948 , 0.2646457729497718 , 0.949661069236870 )
Ponto F     ( -0.1552896503444658 , 0.2720826677165937 , 0.9496610692368700 )
Ponto G     ( -0.1788503468259220 , 0.2572088781829511 , 0.9496610692368700 )
============================================================
************************************************************
Cicle 1024
Ponto C     ( -0.1788503468259220 , 0.2572088781829511 , 0.9496610692368700 )
Ponto D     ( -0.1941693479481317 , 0.2792394929592365 , 0.9403847988399568 )
Ponto medio ( -0.1865098473870268 , 0.2682241855710938 , 0.9450229340384135 )
Ponto F     ( -0.1748548948474560 , 0.2763284637696839 , 0.9450229340384135 )
Ponto G     ( -0.1981647999265981 , 0.2601199073725034 , 0.9450229340384135 )
============================================================
Arquivo codificado com  1024 bytes
----------------------------------------------
```

Figura 5.1: Resultados MME 1024 amostras.

O Modelo de Minimização de entropia da informação é iterativo e utiliza as coordenadas atuais para identificação das coordenadas posteriores, limitando-se à quantidade de iterações realizadas. Desta forma, a entropia da informação é zero uma vez que não há incerteza na recuperação da informação.

Tabela 5.3: Primeiro experimento. Coordenadas resultantes MME.

Experimento com 1024 amostras uniformemente distribuídas MME.			
Coordenada X	**Coordenada Y**	**Coordenada Z**	**Iterações**
-0.1981647999265981	0.2601199073725034	0.9450229340384135	1024

O esforço computacional na utilização do algoritmo LZW varia devido ao processo de busca de *strings* e escrita em tabela, conforme ilustrado na Figura 5.2. Esse esforço computacional varia no processo de encode (Figura 5.3) devido ao Algoritmo de Relação Inteira Borwein et al. (2004), uma vez que as iterações buscam encontrar dois inteiros que atendam a igualdade, conforme Equação 4.16.

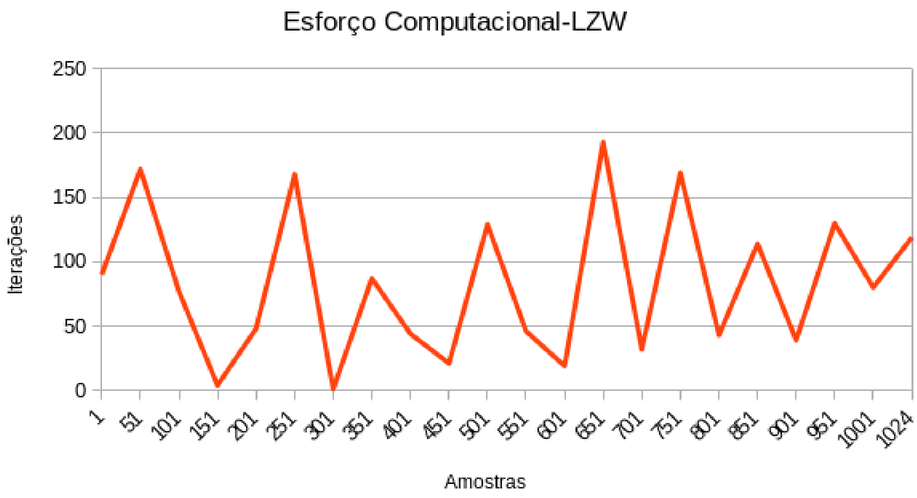


Figura 5.2: Primeiro experimento. Esforço computacional LZW.

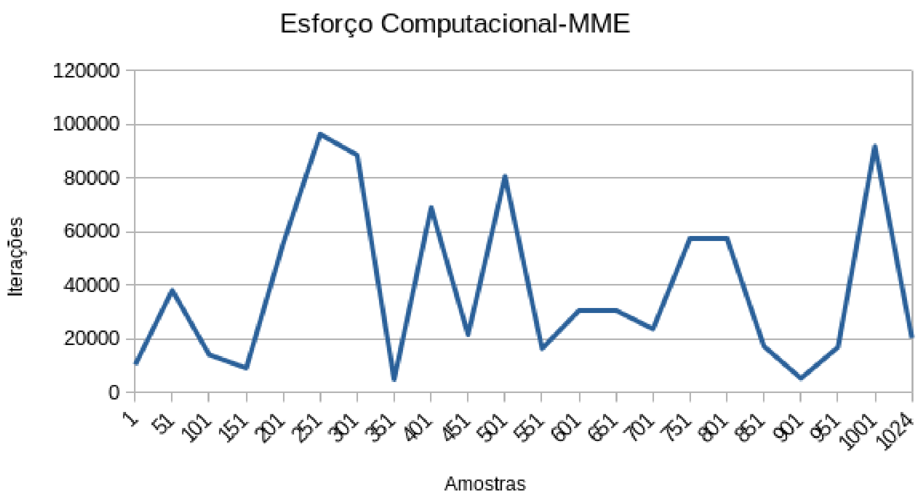


Figura 5.3: Primeiro experimento. Esforço computacional MME.

Observa-se neste experimento a diferença entre do algoritmo LZW quanto ao esforço computacional, onde picos de processamento não ultrapassaram 200 iterações por símbolo codificado, ao passo que no Modelo MME, onde os picos de processamento máximos atingiram cem mil iterações. Também foi observado que a tabela resultante de palavras-código em LZW é substancialmente maior (619 Bytes) em comparação ao MME, com saída de apenas 64 Bytes. Ou seja, apesar de uma melhor compressão, o MME é mais lento, devido à busca pelos ângulos, conforme descrito em seu modelo. Assim, a escolha entre o LZW e o MME deve levar em consideração a disponibilidade de recursos computacionais, bem como a necessidade de minimização de gravação de arquivos resultantes. Para o processo de decodificação, o esforço computacional é igual para todas as amostras, tendo sido identificada uma assimetria entre os dois processos (codificação/decodificação) em ambas as abordagens.

5.3 Compressão de Imagem

Neste experimento foram utilizados 478.537 bytes conforme Figura 5.4 (imagem de teste para amostras de dados) e um conjunto de 256 símbolos passíveis de codificação. Como resultado, obteve-se uma entropia de informação $H(i) = 0,367171390754381$, conforme Tabela 5.4.

Tabela 5.4: Segundo experimento

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
0	1639	0,00342502251654522	0,329825982881556
1	1513	0,00316171999239348	0,308124998616205
2	1610	0,00336442114193887	0,324858122051785
3	1596	0,00333516530592201	0,322454206288305
4	1591	0,00332471679305884	0,321594766356456
5	1723	0,00360055753264638	0,344128997739773
6	1705	0,00356294288633899	0,341074707441055
7	1495	0,00312410534608609	0,304999541494043
8	1699	0,00355040467090319	0,340055335977503
9	1603	0,00334979322393044	0,323656626498041
10	1812	0,0037865410616107	0,359148265679455
11	1817	0,00379698957447386	0,359988058036961
12	1842	0,00384923213878969	0,364180796697857
13	1772	0,00370295295870539	0,352414798205271
14	1636	0,00341875340882732	0,329312790792646
15	1766	0,00369041474326959	0,351402436330559
16	1704	0,00356085318376635	0,340904856621128
17	1746	0,00364862069181693	0,348023429947596
18	1667	0,00348353418857894	0,334607805448936
19	1706	0,00356503258891162	0,341244540515622
20	2087	0,00436120926908473	0,404746412549469
21	1904	0,00397879369829292	0,374534765515249
22	1811	0,00378445135903807	0,358980257140943
23	1986	0,00415014930924882	0,388135170611564
			continua na próxima página...

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
24	2321	0,00485019967108082	0,442674369340925
25	1903	0,00397670399572029	0,374368256300897
26	1755	0,00366742801497063	0,349544835960859
27	1878	0,00392446143140447	0,37020033482601
28	1937	0,00404775388318981	0,380020700537288
29	1794	0,00374892641530331	0,356121547756352
30	1663	0,00347517537828841	0,333925564666268
31	1838	0,00384087332849915	0,363510652215512
32	1555	0,00324948750044406	0,315392675862789
33	1986	0,00415014930924882	0,388135170611564
34	1935	0,00404357447804454	0,379688707012026
35	1824	0,00381161749248229	0,361163068063094
36	1610	0,00336442114193887	0,324858122051785
37	1850	0,00386594975937075	0,365520297577094
38	1544	0,0032265007721451	0,313492591080232
39	1776	0,00371131176899592	0,353089364457252
40	1801	0,00376355433331174	0,35729925116202
41	2310	0,00482721294278185	0,440908030855223
42	1689	0,00352950764517686	0,338354957793871
43	1753	0,00366324860982536	0,34920686667879
44	1890	0,00394953786227606	0,372202186061784
45	1823	0,00380952778990966	0,360995259335875
46	1892	0,00395371726742133	0,372535603334979
47	1735	0,00362563396351797	0,346162026657053
48	1846	0,00385759094908022	0,364850678295907
49	2396	0,00500692736402828	0,454675870576341
50	1619	0,00338322846509256	0,326401549815458
51	2067	0,00431941521763207	0,40146904187648
52	2025	0,00423164770958149	0,394567422517803
53	1881	0,00393073053912237	0,370701014632858
54	1738	0,00363190307123587	0,346669890816385
55	1839	0,00384296303107179	0,363678213012625
56	2070	0,00432568432534997	0,401961019646057
57	2114	0,00441763123854582	0,40916166709682

continua na próxima página...

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
58	1714	0,00358175020949268	0,34260256752044
59	1851	0,00386803946194338	0,365687661476643
60	1921	0,00401431864202768	0,37736299662734
61	1774	0,00370713236385065	0,352752115446504
62	1774	0,00370713236385065	0,352752115446504
63	1907	0,00398506280601082	0,375034197807713
64	1660	0,00346890627057051	0,333413693102063
65	1548	0,00323485958243563	0,314183804367248
66	1840	0,00384505273364442	0,363845757352782
67	1741	0,00363817217895377	0,347177598289012
68	1908	0,00398715250858345	0,37520064348268
69	2031	0,00424418592501729	0,395554970296976
70	1976	0,00412925228352249	0,386482158494172
71	1789	0,00373847790244015	0,35527982517947
72	1925	0,00402267745231821	0,378027799475761
73	1609	0,00336233143936623	0,324686536289175
74	2086	0,0043591195665121	0,404582682312964
75	1801	0,00376355433331174	0,35729925116202
76	1909	0,00398924221115609	0,375367073293535
77	1716	0,00358592961463795	0,342941897540761
78	1825	0,00381370719505493	0,361330860198542
79	1793	0,00374683671273068	0,355953237014823
80	1644	0,00343547102940838	0,330680934031317
81	1911	0,00399342161630135	0,375699885356016
82	2133	0,00445733558742584	0,412262430310151
83	1827	0,00381788660020019	0,361666394730343
84	1800	0,00376146463073911	0,357131058301398
85	1801	0,00376355433331174	0,35729925116202
86	1994	0,00416686692982988	0,389456482399634
87	1632	0,00341039459853679	0,328628275745769
88	1840	0,00384505273364442	0,363845757352782
89	1919	0,00401013923688241	0,377030500692663
90	2343	0,00489617312767874	0,446202315632061
91	1868	0,00390356440567814	0,36853035326266

continua na próxima página...

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
92	1910	0,00399133191372872	0,375533487248558
93	1781	0,00372176028185908	0,353932188977683
94	1825	0,00381370719505493	0,361330860198542
95	1679	0,00350861061945053	0,33665278831384
96	1727	0,00360891634293691	0,344806954152413
97	1900	0,00397043488800239	0,373868633190765
98	2063	0,00431105640734154	0,400812866402486
99	2207	0,00461197357780067	0,424290644122959
100	1664	0,00347726508086104	0,334096152122602
101	1754	0,003665338312398	0,349375859945522
102	1861	0,00388893648766971	0,367360402632832
103	2030	0,00424209622244466	0,395390416311159
104	1956	0,00408745823206983	0,383171530370033
105	2037	0,00425672414045309	0,396541981415969
106	1954	0,00408327882692456	0,382840128112376
107	2222	0,00464331911639016	0,426719552435983
108	1790	0,00374056760501278	0,35544820348748
109	1979	0,00413552139124038	0,386978222642857
110	1868	0,00390356440567814	0,36853035326266
111	1992	0,00416268752468461	0,389126245700422
112	1777	0,00371340147156855	0,353257963400121
113	2221	0,00464122941381753	0,426557720862719
114	1837	0,00383878362592652	0,363343074952524
115	2124	0,00443852826427215	0,410794289415024
116	1796	0,00375310582044858	0,35645811864428
117	1790	0,00374056760501278	0,35544820348748
118	1770	0,00369877355356012	0,352077412656828
119	2077	0,0043403122433584	0,403108456138688
120	1986	0,00415014930924882	0,388135170611564
121	2028	0,00423791681729939	0,39506126358865
122	1814	0,00379072046675597	0,359484232661968
123	1931	0,00403521566775401	0,379024532173355
124	1801	0,00376355433331174	0,35729925116202
125	1787	0,00373429849729488	0,354943017808633

continua na próxima página...

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
126	1768	0,00369459414841486	0,351739958724282
127	1879	0,0039265511339771	0,370367244200087
128	1569	0,00327874333646092	0,317807563803159
129	1722	0,00359846783007375	0,343959464760377
130	1813	0,00378863076418333	0,359316257516738
131	1569	0,00327874333646092	0,317807563803159
132	1723	0,00360055753264638	0,344128997739773
133	1869	0,00390565410825077	0,368697424187859
134	1973	0,00412298317580459	0,385985956462155
135	1631	0,00340830489596416	0,328457100671618
136	1826	0,00381579689762756	0,361498635751276
137	1820	0,00380325868219176	0,360491733512882
138	2044	0,00427135205846152	0,397692818208661
139	2015	0,00421075068385517	0,392920312669673
140	2338	0,00488572461481557	0,445401061317763
141	1995	0,00416895663240251	0,389621577971486
142	2110	0,00440927242825529	0,408508217462616
143	1780	0,00371967057928645	0,353763658093318
144	1618	0,00338113876251993	0,326230132733933
145	1926	0,00402476715489084	0,378193960856075
146	1751	0,0036590692046801	0,348868828351912
147	1584	0,00331008887505041	0,320390751598841
148	2235	0,00467048524983439	0,428822124659192
149	1617	0,0033790490599473	0,326058696956293
150	1855	0,00387639827223391	0,366356953655963
151	1656	0,00346054746027998	0,332730942486253
152	1925	0,00402267745231821	0,378027799475761
153	1697	0,00354622526575792	0,339715403138414
154	1943	0,0040602920986256	0,381016306308546
155	1787	0,00373429849729488	0,354943017808633
156	2374	0,00496095390743036	0,451162903489618
157	1836	0,00383669392335389	0,363175481214733
158	1955	0,0040853685294972	0,383005836983328
159	1773	0,00370504266127802	0,352583465359489

continua na próxima página...

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
160	1685	0,00352114883488633	0,337674305549944
161	1720	0,00359428842492848	0,343620346079014
162	1943	0,0040602920986256	0,381016306308546
163	1960	0,00409581704236036	0,38383414923191
164	1778	0,00371549117414118	0,353426545314063
165	1979	0,00413552139124038	0,386978222642857
166	2092	0,0043716577819479	0,405564846236611
167	1825	0,00381370719505493	0,361330860198542
168	1648	0,00344382983969892	0,331364563657877
169	1676	0,00350234151173263	0,336141786580988
170	1803	0,003767733738457	0,357635586484008
171	1795	0,00375101611787594	0,356289841629717
172	1900	0,00397043488800239	0,373868633190765
173	2306	0,00481885413249132	0,440265332913567
174	1953	0,00408118912435193	0,382674403749281
175	1797	0,00375519552302121	0,356626378809395
176	1788	0,00373638819986751	0,355111429956329
177	1905	0,00398088340086555	0,374701258832296
178	1733	0,00362145455837271	0,345823363369394
179	1964	0,00410417585265089	0,384496520975052
180	2060	0,00430478729962364	0,400320580713629
181	2262	0,00472690721929548	0,433181702238192
182	1951	0,00407700971920667	0,382342908515058
183	1917	0,00400595983173715	0,376697941663821
184	1796	0,00375310582044858	0,35645811864428
185	1750	0,00365697950210746	0,348699783272122
186	1878	0,00392446143140447	0,37020033482601
187	1922	0,00401640834460031	0,377529220954816
188	2079	0,00434449164850367	0,403436163825436
189	1988	0,00415432871439408	0,388465589867288
190	1801	0,00376355433331174	0,35729925116202
191	1978	0,00413343168866775	0,386812883234137
192	1610	0,00336442114193887	0,324858122051785
193	1912	0,00399551131887398	0,375866267624168

continua na próxima página...

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
194	1723	0,0036005575326463 8	0,344128997739773
195	2098	0,00438419599738369	0,406546489081728
196	1946	0,0040665612063435	0,381513898763039
197	1987	0,00415223901182145	0,388300387857378
198	2336	0,00488154520967031	0,44508046892385
199	2173	0,00454092369033115	0,418773653959266
200	1646	0,00343965043455365	0,33102278560292
201	1731	0,00361727515322744	0,345484630243031
202	1708	0,00356921199405689	0,341584153470126
203	1756	0,00366951771754326	0,349713794734595
204	1832	0,00382833511306336	0,362504941336611
205	2026	0,00423373741215413	0,394732051149478
206	2351	0,0049128907482598	0,447483650428847
207	2018	0,00421701979157307	0,39341460305041
208	1707	0,00356712229148425	0,341414355855195
209	2025	0,00423164770958149	0,394567422517803
210	1665	0,00347935478343367	0,334266721397907
211	2019	0,0042191094941457	0,393579336495119
212	1653	0,00345427835256208	0,332218687742897
213	1724	0,00360264723521901	0,34429851315853
214	2313	0,00483348205049975	0,441389916590397
215	1975	0,00412716258094985	0,3863167731475
216	1749	0,00365488979953483	0,348530720901652
217	1729	0,00361309574808218	0,345145827197566
218	1863	0,00389311589281498	0,367694755426831
219	2045	0,00427344176103415	0,397857164155736
220	1733	0,00362145455837271	0,345823363369394
221	1978	0,00413343168866775	0,386812883234137
222	2167	0,00452838547489536	0,417798402087632
223	2125	0,00444061796684478	0,410957473121326
224	1746	0,00364862069181693	0,348023429947596
225	1778	0,00371549117414118	0,353426545314063
226	1947	0,00406865090891613	0,381679731792777
227	2283	0,00477079097332077	0,436565729490967
continua na próxima página...			

Tabela 5.4 – continuando da página anterior.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
228	1836	0,00383669392335389	0,363175481214733
229	1624	0,00339367697795573	0,327258355183629
230	2028	0,00423791681729939	0,39506126358865
231	2356	0,00492333926112296	0,448284065677763
232	1849	0,00386386005679812	0,365352917318064
233	1714	0,00358175020949268	0,34260256752044
234	1549	0,00323694928500826	0,314356558810806
235	2102	0,00439255480767422	0,407200628890096
236	1718	0,00359010901978321	0,343281157033015
237	2059	0,00430269759705101	0,400156456096991
238	1854	0,00387430856966128	0,366189655106394
239	2317	0,00484184086079028	0,442032247565515
240	1672	0,0034939827014421	0,335460198168567
241	2010	0,004200302170992	0,392096194767791
242	1597	0,00333725500849464	0,322626037368381
243	2223	0,0046454088189628	0,426881370377979
244	1709	0,00357130169662952	0,341753933370756
245	1825	0,00381370719505493	0,361330860198542
246	1726	0,00360682664036428	0,344637491354719
247	2188	0,00457226922892065	0,421209591531161
248	1797	0,00375519552302121	0,356626378809395
249	1812	0,0037865410616107	0,359148265679455
250	1747	0,00365071039438957	0,348192544249255
251	2122	0,00443434885912688	0,410467879223918
252	1664	0,00347726508086104	0,334096152122602
253	2044	0,00427135205846152	0,397692818208661
254	1882	0,003932820241695	0,370867875708608
Ocorrências	478.537		0,367171390754381

Observa-se que após a codificação com algoritmo LZW, o conjunto de dados apresentou uma entropia de $0,14452825257157$, conforme Tabela 5.5. A quantidade de Bytes utilizados foi reduzida para a 382.820, caracterizando um ganho de 95.717 Bytes na representação da informação contida nestes dados. A codificação MME (Figura 5.5 e Tabela 5.6) apresenta, como esperado, um conjunto de coordenadas espaciais cuja iteração permitirá a recuperação

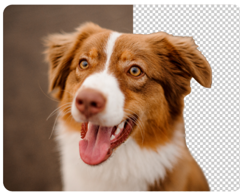


Figura 5.4: Segundo experimento. Imagem de teste.

dos dados originais, de forma direta e sem incerteza. O esforço computacional no processo de codificação apresentou o comportamento ilustrado nas Figuras 5.6 e 5.7.

Tabela 5.5: Codificação arquivo utilizando LZW

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
0	2	0,0032626427406199	0,31647902908981
1	2	0,0032626427406199	0,31647902908981
2	1	0,00163132137030995	0,174571956191618
3	3	0,00489396411092985	0,446032942016868
4	2	0,0032626427406199	0,31647902908981
5	3	0,00489396411092985	0,446032942016868
6	4	0,00652528548123981	0,567551127854691
7	1	0,00163132137030995	0,174571956191618
8	0	0	0
9	4	0,00652528548123981	0,567551127854691
10	1	0,00163132137030995	0,174571956191618
11	0	0	0
12	4	0,00652528548123981	0,567551127854691
13	4	0,00652528548123981	0,567551127854691
14	3	0,00489396411092985	0,446032942016868
15	0	0	0
16	1	0,00163132137030995	0,174571956191618
17	7	0,0114192495921697	0,900615044101755
18	7	0,0114192495921697	0,900615044101755
19	2	0,0032626427406199	0,31647902908981
20	5	0,00815660685154976	0,683084045584213
21	5	0,00815660685154976	0,683084045584213
22	2	0,0032626427406199	0,31647902908981
23	3	0,00489396411092985	0,446032942016868

continua na próxima página...

Tabela 5.5 – continuando da página anterior.

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
24	2	0,0032626427406199	0,31647902908981
25	7	0,0114192495921697	0,900615044101755
26	4	0,00652528548123981	0,567551127854691
27	5	0,00815660685154976	0,683084045584213
28	5	0,00815660685154976	0,683084045584213
29	1	0,00163132137030995	0,174571956191618
30	1	0,00163132137030995	0,174571956191618
31	2	0,0032626427406199	0,31647902908981
32	1	0,00163132137030995	0,174571956191618
33	4	0,00652528548123981	0,567551127854691
34	2	0,0032626427406199	0,31647902908981
35	2	0,0032626427406199	0,31647902908981
36	2	0,0032626427406199	0,31647902908981
37	2	0,0032626427406199	0,31647902908981
38	2	0,0032626427406199	0,31647902908981
39	2	0,0032626427406199	0,31647902908981
40	2	0,0032626427406199	0,31647902908981
41	7	0,0114192495921697	0,900615044101755
42	2	0,0032626427406199	0,31647902908981
43	0	0	0
44	1	0,00163132137030995	0,174571956191618
45	3	0,00489396411092985	0,446032942016868
46	6	0,00978792822185971	0,793839362820932
47	0	0	0
48	3	0,00489396411092985	0,446032942016868
49	7	0,0114192495921697	0,900615044101755
50	2	0,0032626427406199	0,31647902908981
51	2	0,0032626427406199	0,31647902908981
52	3	0,00489396411092985	0,446032942016868
53	3	0,00489396411092985	0,446032942016868
54	3	0,00489396411092985	0,446032942016868
55	3	0,00489396411092985	0,446032942016868
56	3	0,00489396411092985	0,446032942016868
57	5	0,00815660685154976	0,683084045584213

continua na próxima página...

Tabela 5.5 – continuando da página anterior.

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
58	2	0,0032626427406199	0,31647902908981
59	3	0,00489396411092985	0,446032942016868
60	0	0	0
61	3	0,00489396411092985	0,446032942016868
62	1	0,00163132137030995	0,174571956191618
63	0	0	0
64	0	0	0
65	7	0,0114192495921697	0,900615044101755
66	2	0,0032626427406199	0,31647902908981
67	4	0,00652528548123981	0,567551127854691
68	6	0,00978792822185971	0,793839362820932
69	0	0	0
70	2	0,0032626427406199	0,31647902908981
71	6	0,00978792822185971	0,793839362820932
72	2	0,0032626427406199	0,31647902908981
73	3	0,00489396411092985	0,446032942016868
74	8	0,0130505709624796	1,00397821671232
75	1	0,00163132137030995	0,174571956191618
76	4	0,00652528548123981	0,567551127854691
77	4	0,00652528548123981	0,567551127854691
78	2	0,0032626427406199	0,31647902908981
79	1	0,00163132137030995	0,174571956191618
80	1	0,00163132137030995	0,174571956191618
81	6	0,00978792822185971	0,793839362820932
82	4	0,00652528548123981	0,567551127854691
83	4	0,00652528548123981	0,567551127854691
84	4	0,00652528548123981	0,567551127854691
85	7	0,0114192495921697	0,900615044101755
86	4	0,00652528548123981	0,567551127854691
87	2	0,0032626427406199	0,31647902908981
88	4	0,00652528548123981	0,567551127854691
89	3	0,00489396411092985	0,446032942016868
90	2	0,0032626427406199	0,31647902908981
91	1	0,00163132137030995	0,174571956191618
			continua na próxima página...

Tabela 5.5 – continuando da página anterior.

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
92	0	0	0
93	1	0,00163132137030995	0,174571956191618
94	0	0	0
95	3	0,00489396411092985	0,446032942016868
96	4	0,00652528548123981	0,567551127854691
97	5	0,00815660685154976	0,683084045584213
98	4	0,00652528548123981	0,567551127854691
99	6	0,00978792822185971	0,793839362820932
100	3	0,00489396411092985	0,446032942016868
101	2	0,0032626427406199	0,31647902908981
102	10	0,0163132137030995	1,20206822149795
103	5	0,00815660685154976	0,683084045584213
104	6	0,00978792822185971	0,793839362820932
105	6	0,00978792822185971	0,793839362820932
106	0	0	0
107	4	0,00652528548123981	0,567551127854691
108	1	0,00163132137030995	0,174571956191618
109	1	0,00163132137030995	0,174571956191618
110	1	0,00163132137030995	0,174571956191618
111	0	0	0
112	1	0,00163132137030995	0,174571956191618
113	3	0,00489396411092985	0,446032942016868
114	3	0,00489396411092985	0,446032942016868
115	0	0	0
116	4	0,00652528548123981	0,567551127854691
117	0	0	0
118	2	0,0032626427406199	0,31647902908981
119	3	0,00489396411092985	0,446032942016868
120	2	0,0032626427406199	0,31647902908981
121	0	0	0
122	3	0,00489396411092985	0,446032942016868
123	1	0,00163132137030995	0,174571956191618
124	0	0	0
125	3	0,00489396411092985	0,446032942016868

continua na próxima página...

Tabela 5.5 – continuando da página anterior.

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
126	2	0,0032626427406199	0,31647902908981
127	2	0,0032626427406199	0,31647902908981
128	0	0	0
129	3	0,00489396411092985	0,446032942016868
130	2	0,0032626427406199	0,31647902908981
131	4	0,00652528548123981	0,567551127854691
132	3	0,00489396411092985	0,446032942016868
133	2	0,0032626427406199	0,31647902908981
134	3	0,00489396411092985	0,446032942016868
135	3	0,00489396411092985	0,446032942016868
136	7	0,0114192495921697	0,900615044101755
137	2	0,0032626427406199	0,31647902908981
138	4	0,00652528548123981	0,567551127854691
139	1	0,00163132137030995	0,174571956191618
140	4	0,00652528548123981	0,567551127854691
141	3	0,00489396411092985	0,446032942016868
142	2	0,0032626427406199	0,31647902908981
143	1	0,00163132137030995	0,174571956191618
144	4	0,00652528548123981	0,567551127854691
145	4	0,00652528548123981	0,567551127854691
146	0	0	0
147	1	0,00163132137030995	0,174571956191618
148	5	0,00815660685154976	0,683084045584213
149	5	0,00815660685154976	0,683084045584213
150	3	0,00489396411092985	0,446032942016868
151	1	0,00163132137030995	0,174571956191618
152	2	0,0032626427406199	0,31647902908981
153	4	0,00652528548123981	0,567551127854691
154	2	0,0032626427406199	0,31647902908981
155	0	0	0
156	0	0	0
157	1	0,00163132137030995	0,174571956191618
158	0	0	0
159	0	0	0
continua na próxima página...			

Tabela 5.5 – continuando da página anterior.

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
160	3	0,00489396411092985	0,446032942016868
161	5	0,00815660685154976	0,683084045584213
162	1	0,00163132137030995	0,174571956191618
163	5	0,00815660685154976	0,683084045584213
164	1	0,00163132137030995	0,174571956191618
165	5	0,00815660685154976	0,683084045584213
166	3	0,00489396411092985	0,446032942016868
167	3	0,00489396411092985	0,446032942016868
168	2	0,0032626427406199	0,31647902908981
169	1	0,00163132137030995	0,174571956191618
170	0	0	0
171	0	0	0
172	1	0,00163132137030995	0,174571956191618
173	0	0	0
174	0	0	0
175	1	0,00163132137030995	0,174571956191618
176	2	0,0032626427406199	0,31647902908981
177	1	0,00163132137030995	0,174571956191618
178	0	0	0
179	0	0	0
180	3	0,00489396411092985	0,446032942016868
181	0	0	0
182	2	0,0032626427406199	0,31647902908981
183	0	0	0
184	0	0	0
185	4	0,00652528548123981	0,567551127854691
186	0	0	0
187	2	0,0032626427406199	0,31647902908981
188	0	0	0
189	1	0,00163132137030995	0,174571956191618
190	0	0	0
191	2	0,0032626427406199	0,31647902908981
192	4	0,00652528548123981	0,567551127854691
193	1	0,00163132137030995	0,174571956191618

continua na próxima página...

Tabela 5.5 – continuando da página anterior.

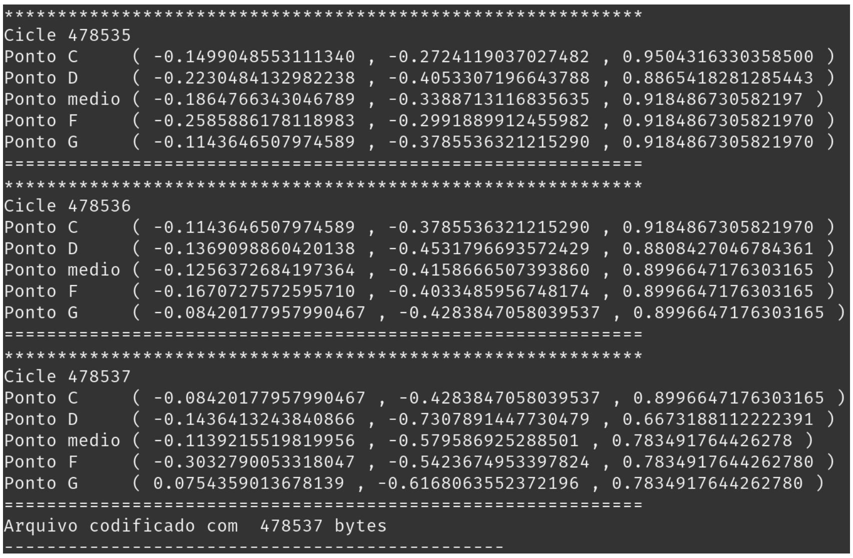
478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
194	0	0	0
195	4	0,00652528548123981	0,567551127854691
196	3	0,00489396411092985	0,446032942016868
197	5	0,00815660685154976	0,683084045584213
198	8	0,0130505709624796	1,00397821671232
199	1	0,00163132137030995	0,174571956191618
200	2	0,0032626427406199	0,31647902908981
201	3	0,00489396411092985	0,446032942016868
202	0	0	0
203	0	0	0
204	1	0,00163132137030995	0,174571956191618
205	0	0	0
206	2	0,0032626427406199	0,31647902908981
207	0	0	0
208	6	0,00978792822185971	0,793839362820932
209	2	0,0032626427406199	0,31647902908981
210	3	0,00489396411092985	0,446032942016868
211	3	0,00489396411092985	0,446032942016868
212	5	0,00815660685154976	0,683084045584213
213	2	0,0032626427406199	0,31647902908981
214	1	0,00163132137030995	0,174571956191618
215	4	0,00652528548123981	0,567551127854691
216	2	0,0032626427406199	0,31647902908981
217	3	0,00489396411092985	0,446032942016868
218	1	0,00163132137030995	0,174571956191618
219	1	0,00163132137030995	0,174571956191618
220	2	0,0032626427406199	0,31647902908981
221	3	0,00489396411092985	0,446032942016868
222	1	0,00163132137030995	0,174571956191618
223	1	0,00163132137030995	0,174571956191618
224	2	0,0032626427406199	0,31647902908981
225	5	0,00815660685154976	0,683084045584213
226	2	0,0032626427406199	0,31647902908981
227	4	0,00652528548123981	0,567551127854691

continua na próxima página...

Tabela 5.5 – continuando da página anterior.

478.537 amostras de imagem codificadas em LZW.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
228	0	0	0
229	3	0,00489396411092985	0,446032942016868
230	4	0,00652528548123981	0,567551127854691
231	1	0,00163132137030995	0,174571956191618
232	1	0,00163132137030995	0,174571956191618
233	2	0,0032626427406199	0,31647902908981
234	0	0	0
235	0	0	0
236	0	0	0
237	2	0,0032626427406199	0,31647902908981
238	1	0,00163132137030995	0,174571956191618
239	0	0	0
240	3	0,00489396411092985	0,446032942016868
241	3	0,00489396411092985	0,446032942016868
242	1	0,00163132137030995	0,174571956191618
243	2	0,0032626427406199	0,31647902908981
244	6	0,00978792822185971	0,793839362820932
245	3	0,00489396411092985	0,446032942016868
246	2	0,0032626427406199	0,31647902908981
247	0	0	0
248	2	0,0032626427406199	0,31647902908981
249	3	0,00489396411092985	0,446032942016868
250	0	0	0
251	3	0,00489396411092985	0,446032942016868
252	0	0	0
253	1	0,00163132137030995	0,174571956191618
254	0	0	0

continua na próxima página...



```
**************************************************************
Cicle 478535
Ponto C     ( -0.1499048553111340 , -0.2724119037027482 , 0.9504316330358500 )
Ponto D     ( -0.2230484132982238 , -0.4053307196643788 , 0.8865418281285443 )
Ponto medio ( -0.1864766343046789 , -0.3388713116835635 , 0.918486730582197 )
Ponto F     ( -0.2585886178118983 , -0.2991889912455982 , 0.9184867305821970 )
Ponto G     ( -0.1143646507974589 , -0.3785536321215290 , 0.9184867305821970 )
==============================================================
**************************************************************
Cicle 478536
Ponto C     ( -0.1143646507974589 , -0.3785536321215290 , 0.9184867305821970 )
Ponto D     ( -0.1369098860420138 , -0.4531796693572429 , 0.8808427046784361 )
Ponto medio ( -0.1256372684197364 , -0.4158666507393860 , 0.8996647176303165 )
Ponto F     ( -0.1670727572595710 , -0.4033485956748174 , 0.8996647176303165 )
Ponto G     ( -0.08420177957990467 , -0.4283847058039537 , 0.8996647176303165 )
==============================================================
**************************************************************
Cicle 478537
Ponto C     ( -0.08420177957990467 , -0.4283847058039537 , 0.8996647176303165 )
Ponto D     ( -0.1436413243840866 , -0.7307891447730479 , 0.6673188112222391 )
Ponto medio ( -0.1139215519819956 , -0.579586925288501 , 0.783491764426278 )
Ponto F     ( -0.3032790053318047 , -0.5423674953397824 , 0.7834917644262780 )
Ponto G     ( 0.0754359013678139 , -0.6168063552372196 , 0.7834917644262780 )
==============================================================
Arquivo codificado com  478537 bytes
-----------------------------------------------
```

Figura 5.5: Segundo experimento. Codificação MME arquivo de imagem.

Tabela 5.6: Segundo experimento. Coordenadas resultantes MME.

Experimento com 478.537 amostras do arquivo de imagem de teste.			
Coordenada X	**Coordenada Y**	**Coordenada Z**	**Iterações**
0,0754359013678139	-0,61680635523722	0,783491764426278	478.537

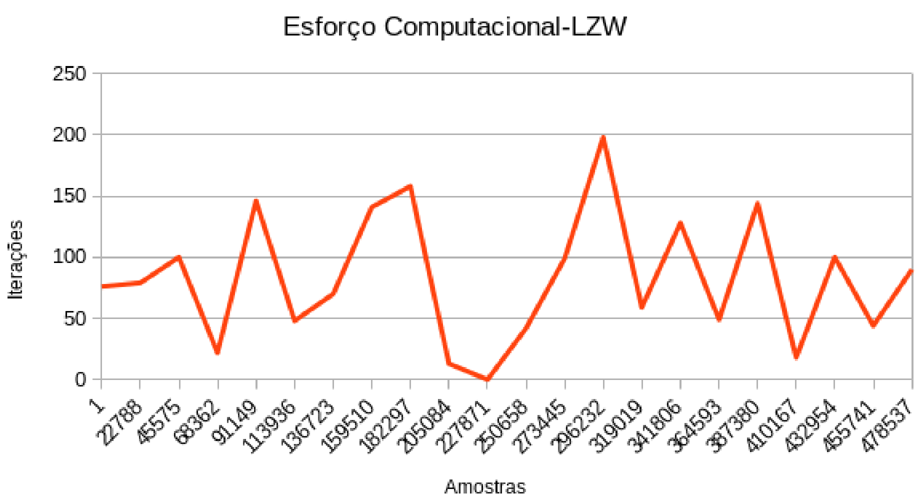
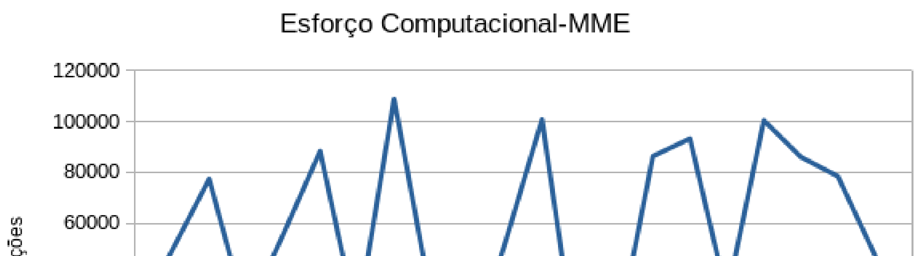


Figura 5.6: Segundo experimento. Esforço computacional LZW.



LZW é substancialmente maior (382.820 Bytes) que a saída do MME (64 Bytes). Desta forma, as características do modelo MME, apresentam-se como um codificador com grande capacidade de compressão, porém o esforço computacional observado proporciona tempos elevados no processo de codificação.

5.4 Dígitos do número π

Neste experimento foi utilizada uma série de 1.000 amostras de dados correspondentes ao primeiros dígitos do número π com um conjunto de 10 símbolos passíveis de codificação, cuja entropia de informação observada foi de $H(i) = 0,46862732937765$, conforme Tabela 5.7.

Tabela 5.7: Terceiro experimento. Símbolos utilizados e entropia calculada.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
0	93	0,093	0,44640493756967
1	116	0,116	0,517752626725263
2	103	0,103	0,478433865459638
3	103	0,103	0,478433865459638
4	93	0,093	0,44640493756967
5	97	0,097	0,459413032704721
6	94	0,094	0,4496822131225
7	95	0,095	0,452942548187283
8	101	0,101	0,472157527246668
9	105	0,105	0,484647739731445
Ocorrências	1.000		0,46862732937765

A aplicação do algoritmo LZW aos conjunto de dados obteve como resultado uma entropia identificada(Tabela 5.8) foi $0,145964742187525$. Também foi observada uma redução na quantidade de símbolos utilizados para 610, caracterizando uma redução de 390 símbolos na representação da informação contida nestes dados.

Tabela 5.8: Terceiro experimento. Entropia identificada após aplicação LZW.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
0	2	0,0032626427406199	0,031647902908981
1	2	0,0032626427406199	0,031647902908981
2	1	0,00163132137030995	0,0174571956191618
3	3	0,00489396411092985	0,0446032942016868
4	2	0,0032626427406199	0,031647902908981
5	3	0,00489396411092985	0,0446032942016868
6	4	0,00652528548123981	0,0567551127854691
7	1	0,00163132137030995	0,0174571956191618
8	0	0	0
9	4	0,00652528548123981	0,0567551127854691
10	1	0,00163132137030995	0,0174571956191618
11	0	0	0
12	4	0,00652528548123981	0,0567551127854691
13	4	0,00652528548123981	0,0567551127854691
14	3	0,00489396411092985	0,0446032942016868
15	0	0	0
16	1	0,00163132137030995	0,0174571956191618
17	7	0,0114192495921697	0,0900615044101755
18	7	0,0114192495921697	0,0900615044101755
19	2	0,0032626427406199	0,031647902908981
20	5	0,00815660685154976	0,0683084045584213
21	5	0,00815660685154976	0,0683084045584213
22	2	0,0032626427406199	0,031647902908981
23	3	0,00489396411092985	0,0446032942016868
24	2	0,0032626427406199	0,031647902908981
25	7	0,0114192495921697	0,0900615044101755
26	4	0,00652528548123981	0,0567551127854691
27	5	0,00815660685154976	0,0683084045584213
28	5	0,00815660685154976	0,0683084045584213
29	1	0,00163132137030995	0,0174571956191618
30	1	0,00163132137030995	0,0174571956191618
31	2	0,0032626427406199	0,031647902908981

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
32	1	0,00163132137030995	0,0174571956191618
33	4	0,00652528548123981	0,0567551127854691
34	2	0,0032626427406199	0,031647902908981
35	2	0,0032626427406199	0,031647902908981
36	2	0,0032626427406199	0,031647902908981
37	2	0,0032626427406199	0,031647902908981
38	2	0,0032626427406199	0,031647902908981
39	2	0,0032626427406199	0,031647902908981
40	2	0,0032626427406199	0,031647902908981
41	7	0,0114192495921697	0,0900615044101755
42	2	0,0032626427406199	0,031647902908981
43	0	0	0
44	1	0,00163132137030995	0,0174571956191618
45	3	0,00489396411092985	0,0446032942016868
46	6	0,00978792822185971	0,0793839362820931
47	0	0	0
48	3	0,00489396411092985	0,0446032942016868
49	7	0,0114192495921697	0,0900615044101755
50	2	0,0032626427406199	0,031647902908981
51	2	0,0032626427406199	0,031647902908981
52	3	0,00489396411092985	0,0446032942016868
53	3	0,00489396411092985	0,0446032942016868
54	3	0,00489396411092985	0,0446032942016868
55	3	0,00489396411092985	0,0446032942016868
56	3	0,00489396411092985	0,0446032942016868
57	5	0,00815660685154976	0,0683084045584213
58	2	0,0032626427406199	0,031647902908981
59	3	0,00489396411092985	0,0446032942016868
60	0	0	0
61	3	0,00489396411092985	0,0446032942016868
62	1	0,00163132137030995	0,0174571956191618
63	0	0	0
64	0	0	0
65	7	0,0114192495921697	0,0900615044101755

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
66	2	0,0032626427406199	0,031647902908981
67	4	0,00652528548123981	0,0567551127854691
68	6	0,00978792822185971	0,0793839362820931
69	0	0	0
70	2	0,0032626427406199	0,031647902908981
71	6	0,00978792822185971	0,0793839362820931
72	2	0,0032626427406199	0,031647902908981
73	3	0,00489396411092985	0,0446032942016868
74	8	0,0130505709624796	0,100397821671232
75	1	0,00163132137030995	0,0174571956191618
76	4	0,00652528548123981	0,0567551127854691
77	4	0,00652528548123981	0,0567551127854691
78	2	0,0032626427406199	0,031647902908981
79	1	0,00163132137030995	0,0174571956191618
80	1	0,00163132137030995	0,0174571956191618
81	6	0,00978792822185971	0,0793839362820931
82	4	0,00652528548123981	0,0567551127854691
83	4	0,00652528548123981	0,0567551127854691
84	4	0,00652528548123981	0,0567551127854691
85	7	0,0114192495921697	0,0900615044101755
86	4	0,00652528548123981	0,0567551127854691
87	2	0,0032626427406199	0,031647902908981
88	4	0,00652528548123981	0,0567551127854691
89	3	0,00489396411092985	0,0446032942016868
90	2	0,0032626427406199	0,031647902908981
91	1	0,00163132137030995	0,0174571956191618
92	0	0	0
93	1	0,00163132137030995	0,0174571956191618
94	0	0	0
95	3	0,00489396411092985	0,0446032942016868
96	4	0,00652528548123981	0,0567551127854691
97	5	0,00815660685154976	0,0683084045584213
98	4	0,00652528548123981	0,0567551127854691
99	6	0,00978792822185971	0,0793839362820931

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
100	3	0,00489396411092985	0,0446032942016868
101	2	0,0032626427406199	0,031647902908981
102	10	0,0163132137030995	0,120206822149795
103	5	0,00815660685154976	0,0683084045584213
104	6	0,00978792822185971	0,0793839362820931
105	6	0,00978792822185971	0,0793839362820931
106	0	0	0
107	4	0,00652528548123981	0,0567551127854691
108	1	0,00163132137030995	0,0174571956191618
109	1	0,00163132137030995	0,0174571956191618
110	1	0,00163132137030995	0,0174571956191618
111	0	0	0
112	1	0,00163132137030995	0,0174571956191618
113	3	0,00489396411092985	0,0446032942016868
114	3	0,00489396411092985	0,0446032942016868
115	0	0	0
116	4	0,00652528548123981	0,0567551127854691
117	0	0	0
118	2	0,0032626427406199	0,031647902908981
119	3	0,00489396411092985	0,0446032942016868
120	2	0,0032626427406199	0,031647902908981
121	0	0	0
122	3	0,00489396411092985	0,0446032942016868
123	1	0,00163132137030995	0,0174571956191618
124	0	0	0
125	3	0,00489396411092985	0,0446032942016868
126	2	0,0032626427406199	0,031647902908981
127	2	0,0032626427406199	0,031647902908981
128	0	0	0
129	3	0,00489396411092985	0,0446032942016868
130	2	0,0032626427406199	0,031647902908981
131	4	0,00652528548123981	0,0567551127854691
132	3	0,00489396411092985	0,0446032942016868
133	2	0,0032626427406199	0,031647902908981

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
134	3	0,00489396411092985	0,0446032942016868
135	3	0,00489396411092985	0,0446032942016868
136	7	0,0114192495921697	0,0900615044101755
137	2	0,0032626427406199	0,031647902908981
138	4	0,00652528548123981	0,0567551127854691
139	1	0,00163132137030995	0,0174571956191618
140	4	0,00652528548123981	0,0567551127854691
141	3	0,00489396411092985	0,0446032942016868
142	2	0,0032626427406199	0,031647902908981
143	1	0,00163132137030995	0,0174571956191618
144	4	0,00652528548123981	0,0567551127854691
145	4	0,00652528548123981	0,0567551127854691
146	0	0	0
147	1	0,00163132137030995	0,0174571956191618
148	5	0,00815660685154976	0,0683084045584213
149	5	0,00815660685154976	0,0683084045584213
150	3	0,00489396411092985	0,0446032942016868
151	1	0,00163132137030995	0,0174571956191618
152	2	0,0032626427406199	0,031647902908981
153	4	0,00652528548123981	0,0567551127854691
154	2	0,0032626427406199	0,031647902908981
155	0	0	0
156	0	0	0
157	1	0,00163132137030995	0,0174571956191618
158	0	0	0
159	0	0	0
160	3	0,00489396411092985	0,0446032942016868
161	5	0,00815660685154976	0,0683084045584213
162	1	0,00163132137030995	0,0174571956191618
163	5	0,00815660685154976	0,0683084045584213
164	1	0,00163132137030995	0,0174571956191618
165	5	0,00815660685154976	0,0683084045584213
166	3	0,00489396411092985	0,0446032942016868
167	3	0,00489396411092985	0,0446032942016868

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
168	2	0,0032626427406199	0,031647902908981
169	1	0,00163132137030995	0,0174571956191618
170	0	0	0
171	0	0	0
172	1	0,00163132137030995	0,0174571956191618
173	0	0	0
174	0	0	0
175	1	0,00163132137030995	0,0174571956191618
176	2	0,0032626427406199	0,031647902908981
177	1	0,00163132137030995	0,0174571956191618
178	0	0	0
179	0	0	0
180	3	0,00489396411092985	0,0446032942016868
181	0	0	0
182	2	0,0032626427406199	0,031647902908981
183	0	0	0
184	0	0	0
185	4	0,00652528548123981	0,0567551127854691
186	0	0	0
187	2	0,0032626427406199	0,031647902908981
188	0	0	0
189	1	0,00163132137030995	0,0174571956191618
190	0	0	0
191	2	0,0032626427406199	0,031647902908981
192	4	0,00652528548123981	0,0567551127854691
193	1	0,00163132137030995	0,0174571956191618
194	0	0	0
195	4	0,00652528548123981	0,0567551127854691
196	3	0,00489396411092985	0,0446032942016868
197	5	0,00815660685154976	0,0683084045584213
198	8	0,0130505709624796	0,100397821671232
199	1	0,00163132137030995	0,0174571956191618
200	2	0,0032626427406199	0,031647902908981
201	3	0,00489396411092985	0,0446032942016868

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
202	0	0	0
203	0	0	0
204	1	0,00163132137030995	0,0174571956191618
205	0	0	0
206	2	0,0032626427406199	0,031647902908981
207	0	0	0
208	6	0,00978792822185971	0,0793839362820931
209	2	0,0032626427406199	0,031647902908981
210	3	0,00489396411092985	0,0446032942016868
211	3	0,00489396411092985	0,0446032942016868
212	5	0,00815660685154976	0,0683084045584213
213	2	0,0032626427406199	0,031647902908981
214	1	0,00163132137030995	0,0174571956191618
215	4	0,00652528548123981	0,0567551127854691
216	2	0,0032626427406199	0,031647902908981
217	3	0,00489396411092985	0,0446032942016868
218	1	0,00163132137030995	0,0174571956191618
219	1	0,00163132137030995	0,0174571956191618
220	2	0,0032626427406199	0,031647902908981
221	3	0,00489396411092985	0,0446032942016868
222	1	0,00163132137030995	0,0174571956191618
223	1	0,00163132137030995	0,0174571956191618
224	2	0,0032626427406199	0,031647902908981
225	5	0,00815660685154976	0,0683084045584213
226	2	0,0032626427406199	0,031647902908981
227	4	0,00652528548123981	0,0567551127854691
228	0	0	0
229	3	0,00489396411092985	0,0446032942016868
230	4	0,00652528548123981	0,0567551127854691
231	1	0,00163132137030995	0,0174571956191618
232	1	0,00163132137030995	0,0174571956191618
233	2	0,0032626427406199	0,031647902908981
234	0	0	0
235	0	0	0

continua na próxima página...

Tabela 5.8 – continuando da página anterior.

Experimento com 1000 primeiros dígitos de π.			
Símbolo	**Frequência**	**Frequência Relativa**	**H(i)**
236	0	0	0
237	2	0,0032626427406199	0,031647902908981
238	1	0,00163132137030995	0,0174571956191618
239	0	0	0
240	3	0,00489396411092985	0,0446032942016868
241	3	0,00489396411092985	0,0446032942016868
242	1	0,00163132137030995	0,0174571956191618
243	2	0,0032626427406199	0,031647902908981
244	6	0,00978792822185971	0,0793839362820931
245	3	0,00489396411092985	0,0446032942016868
246	2	0,0032626427406199	0,031647902908981
247	0	0	0
248	2	0,0032626427406199	0,031647902908981
249	3	0,00489396411092985	0,0446032942016868
250	0	0	0
251	3	0,00489396411092985	0,0446032942016868
252	0	0	0
253	1	0,00163132137030995	0,0174571956191618
254	0	0	0

A codificação MME (Figura 5.8 e Tabela 5.9) apresenta, como esperado, um conjunto de coordenadas espaciais cuja iteração permitirá a recuperação dos dados originais, de forma direta e sem incerteza. O esforço computacional no processo de codificação apresentou o comportamento ilustrado nas Figuras 5.9 e 5.10. Foram observados picos de processamento que não ultrapassaram 180 iterações por símbolo codificado no algoritmo LZW. No Modelo MME, os picos de processamento máximos atingiram 90.000 iterações.

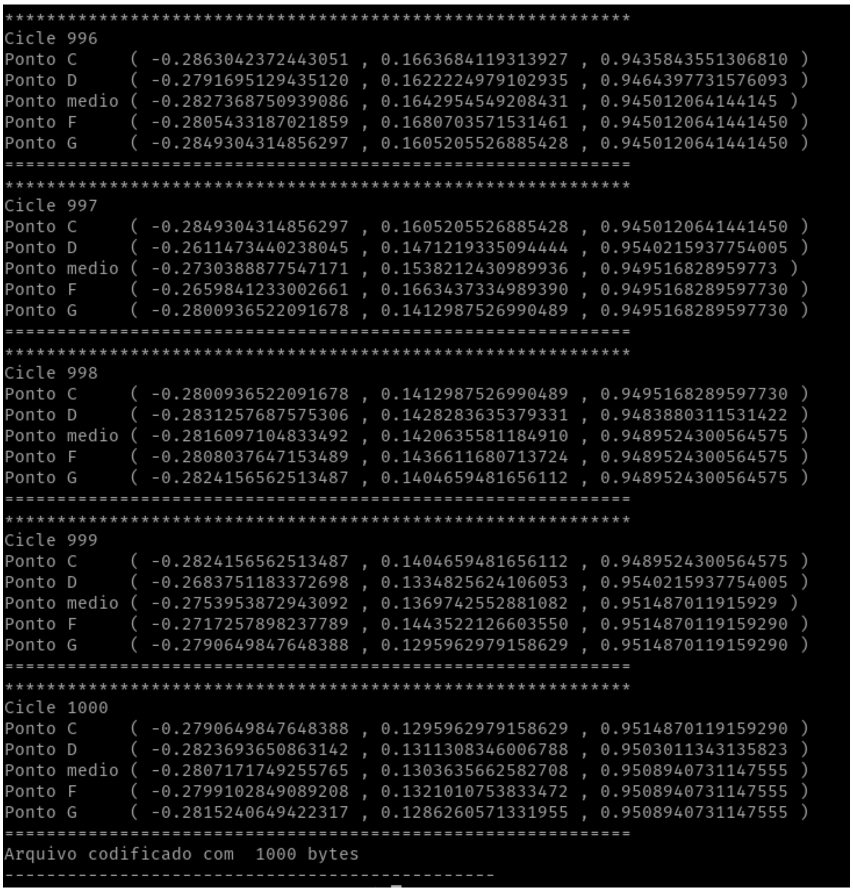


Figura 5.8: Terceiro experimento. Resultados MME 1000 amostras.

Tabela 5.9: Segundo experimento. Coordenadas resultantes MME.

Experimento com 1.000 primeiros dígitos π.			
Coordenada X	**Coordenada Y**	**Coordenada Z**	**Iterações**
-0.2815240649422317	0.1286260571331955	0.9508940731147555	1.000

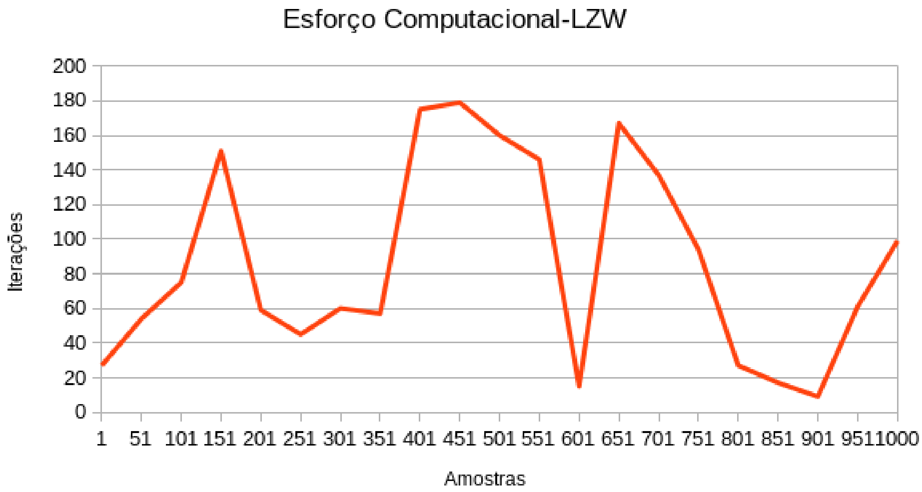


Figura 5.9: Terceiro experimento. Esforço computacional LZW.

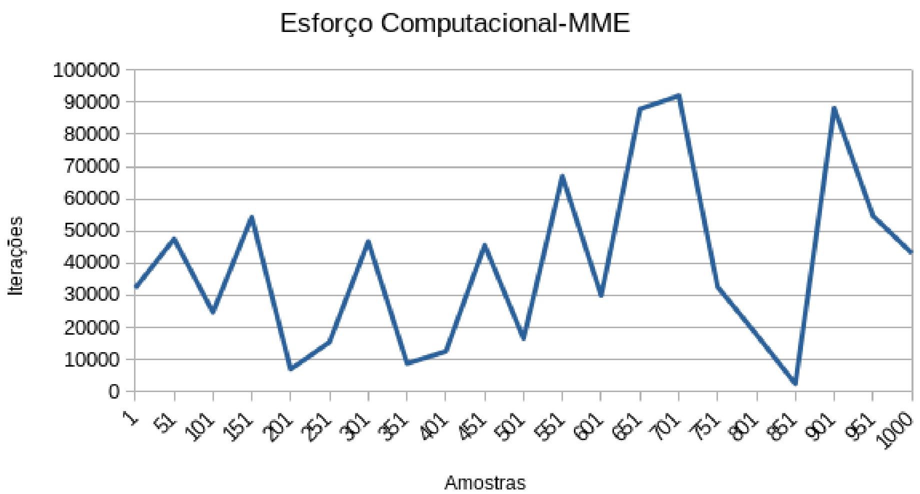


Figura 5.10: Terceiro experimento. Esforço computacional MME.

A partir da análise dos resultados, observa-se que o Modelo de Minimização de Entropia MME apresenta as seguintes características:

1. A precisão utilizada nos experimentos é de 16 dígitos e representa a quantidade de símbolos que podem ser codificados através do MME. Esta precisão pode ser maior ou menor com implicações na quantidade representável de símbolos;

2. O MME apresenta grande sensibilidade a precisão numérica utilizada, não permitindo flutuações na recuperação dos dados codificados. Os valores recuperados devem ser extamente iguais aos valores calculados no processo de codificação;

3. No estágio atual da pesquisa não foi identificada metodologia de correção de possíveis erros de precisão;

4. A Saída obtida a partir da codificação de conjuntos de dados no MME se caracteriza por três coordenadas espaciais e o valor da velocidade angular necessária para a recuperação dos dados durante o processo de decodificação;

5. O MME apresentará como resultado sempre o mesmo bloco de dados independentemente do tamanho do conjunto de entrada. O Tamanho do bloco de dados será calculado em função da precisão utilizada;

6. O MME apresenta grande esforço computacional durante o processo de codificação devido à busca iterativa dos parâmetros A e B que atendam a equações 4.17 e 4.18. Desta forma, o processo de codificação pode ser classificado como lento em relação a técnica clássica utilizada como referência neste trabalho e representa um fator limitante;

7. O decodificação de dados no MME apresenta entropia zero, entretanto no atual estágio da pesquisa não foi obtida uma forma direta de recuperação de uma série numérica a partir das coordenadas calculadas, sendo necessárias a equações intermediárias e passos demonstrados no capítulos referente ao Modelo;

6

Conclusões

Como contribuição aos desafios de aumento da capacidade de expansão das redes de transferência de dados, neste trabalho apresentou-se o Modelo de Minimização de Entropia (MME) da informação em dados digitais. O MME foi apresentado como uma abordagem de compressão de dados sem perda, geométrica, iterativa e não dependente de redundância de dados. O MME tem como resultado as coordenadas, que de um ponto em um espaço tridimensional, permitem a recuperação do conjunto de dados originais. Nesta abordagem, o esforço computacional de decodificação é menor que o necessário ao processo de codificação, diminuindo sensivelmente a necessidade de aumento da infraestrutura da Internet para atendimento às demandas já mencionadas.

Com base nos experimentos realizados neste trabalho, o MME apresentou estabilidade em seu processamento, uma vez que sua normalização de espaço permite o tratamento de valores discretos como identificadores de símbolos, em um contexto de espaço contínuo. Os dados resultantes do MME são invariáveis em quantidade de símbolos, considerando a precisão numérica adotada e independentes do tamanho do conjunto de dados origem. Este fato é muito significativo, pois fornece indícios que a aplicação do modelo em escala comercial pode melhorar sobremaneira o desempenho de diversas soluções atualmente existentes, em diferentes métricas, mas especialmente capacidade de armazenamento e velocidade de transmissão de dados.

Para este momento, os resultados dos experimentos demonstram a viabilidade do MME para utilização em codificação assimétrica, onde o esforço computacional se concentra no emissor da mensagem, apresentando-se como uma possível solução para as maiores demandas como *streaming* multimídia, bem como armazenamento e transferências em geral, com possibilidades que vão desde a redução da demanda de *data centers* até a criação de CDNs (Rede de Distribuição de Conteúdo) virtuais em sua da camada de rede, através de protocolos como o Global Media Transmission Protocol (GMTP) de Sales et al. (2014b, 2020, 2014a). Além disso, a utilização do MME em dispositivos hoje existentes permitirá e multiplicação de suas capacidades sem a alteração do hardware, uma vez que os mesmos datagramas poderiam representar massas de dados limitadas apenas pela capacidade de processamento do receptor final.

Por apresentar uma compressão assimétrica, onde a codificação apresenta um esforço computacional muito maior que a decodificação, por enquanto, o MME limita-se ao uso em ambientes onde o recurso computacional disponível (hardware) atende às demandas de seu processamento. Apesar disso, vale salientar que, na maioria dos casos, os processos de codificação necessitam apenas de um evento de compressão, o que diminui sensivelmente o impacto desta limitação.

Por último, mas não por fim, a proposta de minimização da entropia apresentada neste trabalho possibilitará contribuições futuras em diferentes áreas, possibilitando uma abordagem determinística aos problemas hoje limitados ao tratamento estocástico, uma vez que conjuntos de dados que descrevem diversos eventos físicos e biológicos, hoje são tratados estatisticamente e poderão ser generalizados em equações determinísticas. Nesse contexto, acredita-se que se abre uma nova linha de pesquisa a ser aplicada em diferentes áreas de conhecimento, tais como:

1. O mapeamento dos estados de Sistemas Termodinâmicos através do MME permitirá a previsão das suas transições de forma iterativa e sem incerteza;

2. A projeção de Sistemas biológicos modelados por espirais no espaço métrico mapeado pelo MME, permitirá a identificação de padrões de evolução destas espirais.

3. A aplicação da geometria do MME no mapeamento de séries de dados, permitirá a identificação de valores intermediários no espaço métrico estudado, e possibilitará a criação de Interpolador Numérico.

4. A projeção de polítopos no espaço métrico do MME possibilitará a melhoraria em técnicas de Otimização matemática.

5. A informação representada através da geometria do MME, e possibilitará funções de ativação aplicáveis às Redes Neurais Artificiais;

6.1 Trabalhos futuros

Ao longo deste trabalho, procurou-se contemplar diversos aspectos do MME, seja em desempenho e exigência computacional, seja em saída de dados resultantes. O MME apresenta-se com um grande potencial de desenvolvimento na academia, onde os processos de otimização e mapeamento de estados térmicos podem contribuir significativamente para a solução de problemas hoje estudados. No mercado, sua contribuição torna-se mais evidente, onde a demanda crescente por transmissão de dados, conforme já descrito neste trabalho, representa um ambiente natural para a aplicação do modelo MME. Desta forma, diversos tópicos poder ser investigados futuramente como:

1. Geração de séries numéricas a partir das saídas de dados do MME, aprofundamento matemático e criação de modelo analítico;
2. Desenvolvimento de API para integração em ambiente de *streaming* de vídeo;
3. Desenvolvimento de protocolo de comunicação com informações condensadas;
4. Desenvolvimento de sistema de arquivos que permitirá uma maior capacidade de armazenamento de dados em dispositivos;
5. Aplicação do Modelo MME em estudos de otimização.
6. Aplicação do Modelo MME no mapeamento de estados de sistemas térmicos e previsão iterativa de suas transições.

Referências Bibliográficas

Jürgen Abel and William Teahan. Universal text preprocessing for data compression. *IEEE Transactions on Computers*, 54(5):497–507, 2005.

Luis Aguiar and Bertin Martens. Digital music consumption on the internet: Evidence from clickstream data. *Information Economics and Policy*, 34:27–43, 2016.

Asad Ali. Código fonte do algoritmo lzw, 2014. https://github.com/asad82/LZW-Compression.

Mariana Olivieri Caixeta Altoé and MS Pinho. Transmissão de informação embutida em arquivos comprimidos com o algoritmo lzw. *Anais do XXII Simpósio Brasileiro de Telecomunicações SBT'05*, 2005.

FN Auristin and SD Mali. Advanced audio compression for lossless audio coding using ieee 1857.2. *Int. J. Eng. Comput. Sci*, 5(09):18124–18127, 2016.

David H Bailey and Jonathan M Borwein. Pslq: an algorithm to discover integer relations. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2009.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pages 675–685. PMLR, 2019.

Jonathan M Borwein, David H Bailey, and Roland Girgensohn. *Experimentation in mathematics: Computational paths to discovery*. AK Peters/CRC Press, 2004.

Aura Conci and Felipe R Aquino. Fractal coding based on image local fractal dimension. *Computational & Applied Mathematics*, 24:83–98, 2005.

J Brian Connell. A huffman-shannon-fano code. *Proceedings of the IEEE*, 61(7):1046–1047, 1973.

Leandro de Sales, Wendell Soares, Thiago de Sales, Hyggo de Almeida, and Angelo Perkusich. Global media transmission protocol (gmtp). In *Anais do I Workshop Pré-IETF*, pages 36–50. SBC, 2014a.

Leandro Melo de Sales, Thiago Sales, Hyggo Almeida, Angelo Perkusich, and Kyller Gorgônio. Generalized connections and incentives for supporting ce devices in live streaming systems. *IEEE Transactions on Consumer Electronics*, 60(4):605–613, 2014b. **DOI** 10.1109/TCE.2014.7027333.

Leandro Melo de Sales, Wendell Soares, Rafael Amorim, Thiago Sales, Karan Verma, and Eduardo Setton. A network of cooperative routers to distribute live multimedia content over the internet. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 751–756, 2020. **DOI** 10.1109/ICNC47757.2020.9049815.

Herman do Lago Mendes. Como medir informação? *REAMEC-Rede Amazônica de Educação em Ciências e Matemática*, 5(2):177–200, 2017.

G. W. Drost and Nikolaos G. Bourbakis. A hybrid system for real-time lossless image compression. *Microprocess. Microsystems*, 25:19–31, 2001.

V Granger, C McFadden, M Lambert, S Carrington, J Oliver, N Barton, D Reingold, and K Still. Net benefits: The internet-a real or virtual threat. *Merrill Lynch report*, 1998.

Bo Hang, Yi Wang, and Changqing Kang. A scalable variable bit rate audio codec based on audio attention analysis. *Revista Técnica de la Facultad de Ingeniería. Universidad del Zulia*, 39(6):114–120, 2016.

Uthayakumar Jayasankar, Vengattaraman Thirumal, and Dhavachelvan Ponnurangam. A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. *Journal of King Saud University - Computer and Information Sciences*, 33(2):119–140, 2021. ISSN 1319-1578. **DOI** https://doi.org/10.1016/j.jksuci.2018.05.006. URL `https://www.sciencedirect.com/science/article/pii/S1319157818301101`.

S Piramu Kailasam. Efficient haar wavelet transform with embedded zerotrees of wavelet compression for color images. *International Journal of Computer and Information Engineering*, 14(11):406–412, 2020.

G. G. Langdon. An introduction to arithmetic coding. *IBM Journal of Research and Development*, 28(2):135–149, 1984. **DOI** 10.1147/rd.282.0135.

Steffen Lange, Johanna Pohl, and Tilman Santarius. Digitalization and energy consumption. does ict reduce energy demand? *Ecological Economics*, 176:106760, 2020.

Jan Lansky and Michal Zemlicka. Compression of small text files using syllables. In *Proceedings. DCC 2006. Data Compression Conference*, pages 458–458. IEEE Computer Society, 2006.

Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking: An overview. In *International conference on runtime verification*, pages 122–135. Springer, 2010.

Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

Chun-Wang Ma and Yu-Gang Ma. Shannon information entropy in heavy-ion collisions. *Progress in Particle and Nuclear Physics*, 99:120–158, 2018.

M Narasimha and A Peterson. On the computation of the discrete cosine transform. *IEEE Transactions on Communications*, 26(6):934–936, 1978.

Jyoti Neeli and Shamshekhar Patil. Insight to security paradigm, research trend & statistics in internet of things (iot). *Global Transitions Proceedings*, 2(1):84–90, 2021.

Kim-Khoa Nguyen and Brigitte Jaumard. Distributed control plane architecture of next generation ip routers. In *2009 IEEE International Conference on Cluster Computing and Workshops*, pages 1–8. IEEE, 2009.

José Octávio de Carvalho Pineda et al. A entropia segundo claude shannon: o desenvolvimento do conceito fundamental da teoria da informação. *Pontifícia Universidade Católica de São Paulo*, 2006.

Srinivasa Ramanujan et al. *Notebooks of Srinivasa Ramanujan: Volume II*, volume 2. Springer, 2013.

YV Ramana Rao and C Eswaran. New bit rate reduction techniques for block truncation coding. *IEEE transactions on communications*, 44(10):1247–1250, 1996.

Mohammad Salahuddin and Khorshed Alam. Information and communication technology, electricity consumption and economic growth in oecd countries: A panel data analysis. *International Journal of Electrical Power & Energy Systems*, 76:185–193, 2016. ISSN 0142-0615. **DOI** https://doi.org/10.1016/j.ijepes.2015.11.005. URL https://www.sciencedirect.com/science/article/pii/S014206151500424X.

David Salomon. Data compression. In *Handbook of massive data sets*, pages 245–309. Springer, 2002.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Andrei B Shidlovskii. Transcendental numbers. In *Transcendental Numbers*. de Gruyter, 2011.

Raymond J. Solomonoff. A preliminary report on a general theory of inductive inference, 1960.

Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

John Watkinson. *The MPEG handbook*. Routledge, 2012.

Terry A. Welch. A technique for high-performance data compression. *Computer*, 17(06):8–19, 1984.

Serge Winitzki. Uniform approximations for transcendental functions. In *International conference on computational science and its applications*, pages 780–789. Springer, 2003.

Kenjiro Yanagi, Ken Kuriyama, and Shigeru Furuichi. Generalized shannon inequalities based on tsallis relative operator entropy. *Linear algebra and its applications*, 394:109–118, 2005.

Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.

Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.