**Multimédia II**

**Universidade Fernando Pessoa**

**Ano Letivo**

****

**Título do trabalho: Algoritmo Adaptive Huffman**

**Elaborado por:**

Carlos

Pedro Marques 38688

Sérgio Sousa 39836

***Índice***

**Resumo**

[**Resumo 2**](#_g26t5yalivwm)

[**1 - Introdução 3**](#_nljnqrcdk9i5)

[1.1 - Objetivos do trabalho 3](#_q3wbwyk1qy0q)

[1.2 - Justificação do interesse do codec 3](#_6alyak62hj2u)

[1.3 Áreas de Aplicação 4](#_6lgfv4gy9ihn)

[**2. Algoritmo Huffman Adaptativo 4**](#_dloobzk6rkhc)

[2.1 - Codificação 4](#_8v9r0t83py21)

[2.2 - Descodificação 4](#_wm85k2iv9toq)

[2.3 - Exemplo de aplicação para apresentação passo a passo 4](#_6cuakyb0m6pa)

[2.3.1 - Codificação 4](#_3alhdzzcy5rj)

[2.3.2 - Descodificação 4](#_gt947ygh9xta)

[**3 - Implementação do algoritmo de Huffman Adaptativo 4**](#_qgwb1s908mo4)

[**4 - Teste do codec com o Corpus Silesia 4**](#_6zx2xv2z7vcl)

[**5 - Conclusão 4**](#_93evqhpm4hc)

[**Bibliografia 5**](#_ywea0taxuqay)

# 

# **Resumo**

Neste relatório pretende-se descrever de forma detalhada o algoritmo de Huffman Adaptativo, tanto na vertente de codificação e descodificação.

Começamos por fazer uma breve introdução sobre a origem do algoritmo, explicar a sua utilidade, as áreas de aplicação do mesmo e o tipo de dados com o qual o algoritmo de codificação funciona. É feita uma introdução sobre os processos de codificação e decodificação , com exemplos de aplicação dos mesmos, seguido do pseudo-código da codificação e descodificação respetivamente.

De seguida é apresentada a realização dos testes perante os ficheiros do Corpus Silesia.

# **1 - Introdução**

O algoritmo Huffman adaptativo foi desenvolvido independentemente por Faller (1973) e Gallager (1978), tendo sido aperfeiçoado por Knuth (1985). Ele foi criado para ser uma técnica de codificação relativamente rápida e de acesso aleatório que não exige conhecimento prévio das estatísticas do símbolo e se adapta automaticamente às mudanças nas estatísticas. O algoritmo é uma variante da codificação Huffman, que atribui códigos de bit de comprimento variável aos símbolos com base em sua probabilidade, com símbolos menos prováveis tendo códigos mais longos. A codificação Huffman adaptativa é capaz de lidar com casos em que as frequências de símbolos são desconhecidas e requer a atualização constante de todas as frequências na árvore. A ideia principal do algoritmo é inicializar a árvore Huffman com um único nó, ou código de escape, e modificar e atualizá-lo à medida que os símbolos são lidos e processados. O processo de codificação envolve a adição de símbolos à árvore e a atribuição de códigos, modificando a árvore para manter a melhor árvore Huffman possível. O decodificador espelha as operações do codificador, e os dois devem trabalhar em sincronia.

## 

## 

## 

## 

## 

## 

## 

## **1.1 - Objetivos do trabalho**

O objectivo deste trabalho é de perceber a implementação e a diferença entre outros algoritmos de compressão para observar quais são os melhores para cada tipo de ficheiros e tamanho de ficheiros também. Ajudando na aprendizagem do mesmo.

## **1.2 - Justificação do interesse do codec**

Devido a durante as aulas de multimédia foi falado do Algoritmo de Huffman, foi despertado um interesse sobre o algoritmo de Huffman Adaptativo, de como consegue expandir perante o original e quais as suas melhorias, em termos de performance como um algoritmo de codec e decodec como nas suas aplicações no âmbito de multimédia.

## **1.3 Áreas de Aplicação**

# 

# O algoritmo de Huffman adaptativo é um algoritmo de compressão de dados que é usado em uma ampla variedade de áreas, incluindo:

# 1. Compressão de arquivos: o algoritmo é frequentemente usado em programas de compressão de arquivos, como o WinZip, para compactar arquivos e reduzir seu tamanho.

# 2. Comunicação de dados: o algoritmo é usado em sistemas de comunicação de dados para compactar dados antes de serem transmitidos, reduzindo assim o tempo e o uso de largura de banda necessários para transmitir os dados.

# 3. Armazenamento de dados: o algoritmo é usado em dispositivos de armazenamento de dados, como discos rígidos e unidades USB, para compactar dados e permitir que mais dados sejam armazenados em um espaço limitado.

# 4. Transmissão de vídeo: o algoritmo é usado em sistemas de transmissão de vídeo, como o HDTV, para comprimir os dados de vídeo e permitir que sejam transmitidos em velocidades mais altas.

# 5. Processamento de imagens: o algoritmo é usado em softwares de processamento de imagens para comprimir imagens e reduzir seu tamanho sem perda significativa de qualidade.

# 6. Reconhecimento de fala: o algoritmo é usado em sistemas de reconhecimento de fala para compactar os dados de áudio antes de serem processados, permitindo que o processamento seja realizado mais rapidamente.

# 

# 2. Algoritmo Huffman Adaptativo

## 2.1 - Codificação

Vai ser explicado em detalhe como funciona a parte de codificação do algoritmo de Huffman Adaptativo:

1. Inicialização: A codificação começa com uma tabela de códigos vazia. Cada símbolo de entrada é inicialmente codificado usando um código fixo de comprimento fixo (por exemplo, 8 bits).

2. Contagem de frequências: A cada novo símbolo de entrada, o algoritmo de Huffman Adaptativo atualiza a tabela de códigos contando a frequência de ocorrência de cada símbolo. Isso é feito mantendo uma contagem para cada símbolo na tabela de códigos. Se o símbolo ainda não foi visto antes, é adicionado à tabela de códigos com uma frequência inicial de 1.

3. Construção da árvore de Huffman: A partir da tabela de códigos atualizada, o algoritmo constrói uma árvore de Huffman. A árvore de Huffman é uma estrutura de árvore binária que é usada para gerar os códigos de Huffman para cada símbolo. Os símbolos com frequências mais altas são colocados mais perto da raiz da árvore, enquanto os símbolos com frequências mais baixas são colocados nas folhas.

4. Geração dos códigos de Huffman: Com a árvore de Huffman construída, o algoritmo pode gerar os códigos de Huffman para cada símbolo de entrada. Os códigos de Huffman são gerados percorrendo a árvore de Huffman da raiz até a folha correspondente ao símbolo de entrada. O caminho da raiz à folha é registrado como o código de Huffman para esse símbolo.

5. Atualização da tabela de códigos: Após a geração dos códigos de Huffman para um símbolo, a tabela de códigos é atualizada com o novo código de Huffman e a frequência do símbolo é atualizada. Isso garante que a tabela de códigos reflita as frequências atualizadas de todos os símbolos de entrada.

6. Saída dos códigos de Huffman: Os códigos de Huffman gerados são então usados para representar os símbolos de entrada em um formato mais compacto. Os códigos de Huffman são de comprimento variável, o que significa que os símbolos mais comuns são representados por códigos mais curtos, enquanto os símbolos menos comuns são representados por códigos mais longos.

Em resumo, a parte de codificação do algoritmo de Huffman Adaptativo funciona criando uma tabela de códigos Huffman, construindo uma árvore de Huffman a partir da tabela, gerando os códigos de Huffman para cada símbolo de entrada, atualizando a tabela de códigos e, finalmente, usando os códigos de Huffman para representar os símbolos de entrada em um formato mais compacto.

## 2.2 - Descodificação

Após a etapa de codificação, a parte de descodificação do algoritmo de Huffman Adaptativo é responsável por reconstruir os símbolos originais a partir dos códigos de Huffman gerados na etapa de codificação. A seguir, vou explicar em detalhes como funciona a parte de descodificação do algoritmo de Huffman Adaptativo:

1. Inicialização: A descodificação começa com a mesma tabela de códigos vazia usada na etapa de codificação.

2. Leitura do código de Huffman: O algoritmo de descodificação lê os códigos de Huffman gerados na etapa de codificação. Para cada código de Huffman, o algoritmo começa na raiz da árvore de Huffman.

3. Descida pela árvore de Huffman: O algoritmo desce pela árvore de Huffman a partir da raiz, seguindo o caminho correspondente ao código de Huffman lido. Quando alcança uma folha, o algoritmo sabe que encontrou um símbolo correspondente ao código de Huffman.

4. Saída do símbolo: Quando um símbolo é encontrado, ele é emitido como saída e adicionado à sequência de saída. A tabela de códigos é atualizada com o novo símbolo e a frequência é incrementada.

5. Atualização da tabela de códigos: Após a emissão do símbolo, a tabela de códigos é atualizada com o novo símbolo e a frequência é incrementada. Isso garante que a tabela de códigos reflita as frequências atualizadas de todos os símbolos de entrada.

6. Repetição do processo: O processo é repetido até que todos os códigos de Huffman tenham sido lidos e os símbolos correspondentes tenham sido emitidos.

Em resumo, a parte de descodificação do algoritmo de Huffman Adaptativo funciona reconstruindo os símbolos originais a partir dos códigos de Huffman gerados na etapa de codificação. Isso é feito seguindo o caminho correspondente ao código de Huffman na árvore de Huffman, emitindo o símbolo correspondente, atualizando a tabela de códigos e repetindo o processo até que todos os códigos de Huffman tenham sido lidos e os símbolos correspondentes tenham sido emitidos.

## 2.3 - Exemplo de aplicação para apresentação passo a passo

### 2.3.1 - Codificação

### 2.3.2 - Descodificação

# 3 - Implementação do algoritmo de Huffman Adaptativo

# 4 - Teste do codec com o Corpus Silesia

# 5 - Conclusão

# Bibliografia

## 