

TP1 Introdução a Computação Visual

Universidade Federal de Minas Gerais

Compressor de Imagens

Alunos: Gabriel Alves Reis, Samuel Brísio Jesus.

Introdução:

Neste trabalho realizamos a implementação de um conversor baseado no método de compressão JPEG.

Pré-Processamento da Imagem de Entrada:

Para realizar a Transformada Discreta do Cosseno (DCT) é necessário que a largura e altura da imagem sejam múltiplos de oito, para isso realizamos um padding à direita e embaixo, simplesmente replicando o valor do tom de cinza. Além disso, antes de realizar a DCT centralizamos a imagem subtraindo por 128, assim, temos a imagem pronta para realizar a DCT e quantizá-la no domínio das frequências.

Quantização:

Na quantização, utilizamos a matriz com fator de qualidade 98 apresentada no artigo Determining JPEG Image Standard Quality Factor from the Quantization Tables.¹

Para quantizar, dividimos o resultado da DCT pela matriz de quantização, para garantir que os valores estão entre 0 e 255 multiplicamos a matriz por um fator calculado em função da diferença entre o valor máximo e mínimo, chamaremos este valor de α . Por fim, truncamos os valores decimais, ficando com uma matriz apenas de inteiros.

Uma característica marcante desta matriz é a grande quantidade de zeros presente, a qual terá grande importância na codificação futura. No entanto, para evitar trabalhar com representações de números negativos, decidimos somar todos os elementos desta matriz quantizada pelo absoluto de seu valor mínimo, vamos chamar este valor de β . Logo, a quantidade de massa de zeros será substituída por uma quantidade massiva de β .

Aplicando a quantização proposta nas imagens abaixo temos: 88% de repetição de β em A e 81% em B.

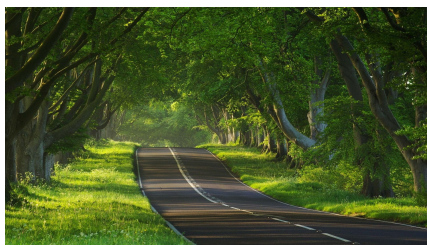


Imagem A (1920 x 1080)

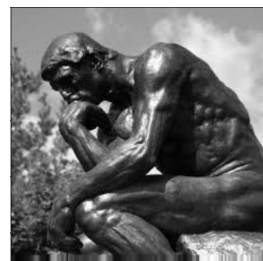
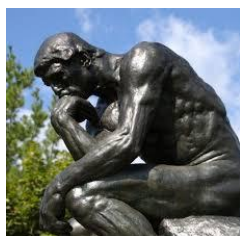


Imagem B (277 x 222) → (280 x 224)

¹ COGRANNE, Rémi. **Determining JPEG Image Standard Quality Factor from the Quantization Tables**. ArXi, França, 2018.

Codificação:

Para uma codificação eficiente nos aproveitamos de dois fatos: O primeiro é que os valores da matriz após a quantização estão entre 0 e 255 inclusos, logo podemos representar com 1 Byte cada. O segundo é a alta repetição de β .

Para codificar realizamos os seguintes passos:

Salvamos nos 2 primeiros bytes do arquivo de saída da compressão a altura da imagem, nos 2 bytes seguintes a largura, no 5º Byte o valor de β e no 6º byte α .

Depois, iteramos por todos os valores da quantização da DCT. Se o valor for β então adicionamos um bit '0' no arquivo de saída, caso contrário, adicionamos um bit '1' seguido de 1 Byte que representa o número em si.

Descompressão:

No final da codificação temos um arquivo comprimido, para descomprimir basta realizar os seguintes passos:

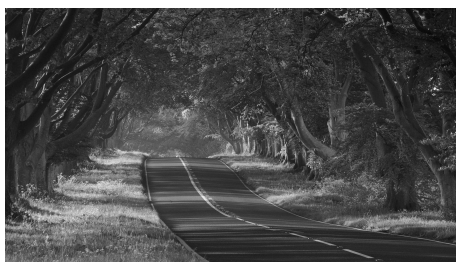
1 - Identificar as dimensões da matriz (M x N), juntamente com o valor de β e α no header (6 primeiros Bytes).

2 - Iterar pelo resto do arquivo, se encontrar um bit '0' insira β em uma pilha, caso encontre um bit '1' leia e insira o próximo byte na pilha. Faça isso até que sua pilha tenha o tamanho M*N.

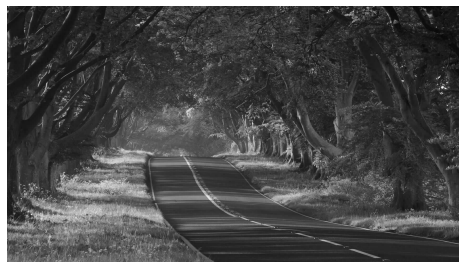
3 - Construa uma matriz M x N com os valores da pilha, subtraia β em cada elemento da matriz, multiplique bloco a bloco pela matriz de quantização e multiplique os valores por $255/\alpha$, por fim, realize a DCT inversa e some 128 para que os tons de cinza fiquem entre 0 e 255.

Exemplos e Métricas:

Entrada



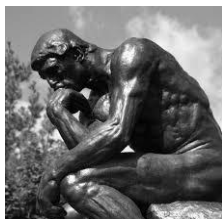
Saída



PSNR: 33.68

Taxa de Compressão: 3.32

Entrada



Saída



PSNR: 34.40

Taxa de Compressão: 3.85