Compressão de Imagens via Álgebra Linear

Prof. Alexandre Lymberopoulos

Instituto de Matemática e Estatística
Universidade de São Paulo

Transferência de Imagens Digitais

- Imagens digitais ocupam muito espaço em disco e requerem muita memória para serem salvas e visualizadas.

Transferência de Imagens Digitais

- Imagens digitais ocupam muito espaço em disco e requerem muita memória para serem salvas e visualizadas.
- Apresentaremos uma maneira de comprimir imagens digitais de modo que utilizem menos espaço ao serem salvas e transmitidas.

Transferência de Imagens Digitais

- Imagens digitais ocupam muito espaço em disco e requerem muita memória para serem salvas e visualizadas.
- Apresentaremos uma maneira de comprimir imagens digitais de modo que utilizem menos espaço ao serem salvas e transmitidas.
- Usaremos as wavelets de Haar ("ondaletas de Haar").

Transferência de Imagens Digitais

- Imagens digitais ocupam muito espaço em disco e requerem muita memória para serem salvas e visualizadas.
- Apresentaremos uma maneira de comprimir imagens digitais de modo que utilizem menos espaço ao serem salvas e transmitidas.
- Usaremos as wavelets de Haar (“ondaletas de Haar”).
- Aqui a palavra wavelet será um sinônimo para uma base ortonormal num certo espaço vetorial.

Modelando o Problema

- A idéia básica é tratar a imagem como uma matriz.

Modelando o Problema

- A idéia básica é tratar a imagem como uma matriz.
- Uma imagem digital é formada de pequenos quadradinhos chamados *pixels* ("picture elements").

Modelando o Problema

- A idéia básica é tratar a imagem como uma matriz.
- Uma imagem digital é formada de pequenos quadradinhos chamados *pixels* ("picture elements").
- A matriz correspondente a uma imagem digital associa um inteiro a cada pixel.

Modelando o Problema

- A idéia básica é tratar a imagem como uma matriz.
- Uma imagem digital é formada de pequenos quadradinhos chamados *pixels* ("picture elements").
- A matriz correspondente a uma imagem digital associa um inteiro a cada pixel.
- Exemplo: uma imagem em tons de cinza com 256×256 pixels é representada por um matriz 256×256, onde cada entrada tem um inteiro entre 0 (preto) e 255 (branco).

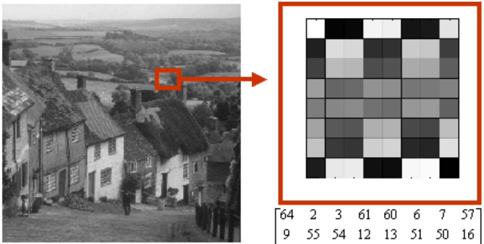
Modelando o Problema

- A idéia básica é tratar a imagem como uma matriz.
- Uma imagem digital é formada de pequenos quadradinhos chamados *pixels* ("picture elements").
- A matriz correspondente a uma imagem digital associa um inteiro a cada pixel.
- Exemplo: uma imagem em tons de cinza com 256×256 pixels é representada por um matriz 256×256, onde cada entrada tem um inteiro entre 0 (preto) e 255 (branco).
- O algoritmo de compressão JPEG subdivide a imagem em blocos de 8×8 pixels e constrói uma matriz para cada bloco.

Modelando o Problema

- A idéia básica é tratar a imagem como uma matriz.
- Uma imagem digital é formada de pequenos quadradinhos chamados *pixels* ("picture elements").
- A matriz correspondente a uma imagem digital associa um inteiro a cada pixel.
- Exemplo: uma imagem em tons de cinza com 256×256 pixels é representada por um matriz 256×256, onde cada entrada tem um inteiro entre 0 (preto) e 255 (branco).
- O algoritmo de compressão JPEG subdivide a imagem em blocos de 8×8 pixels e constrói uma matriz para cada bloco.
- Usamos algumas técnicas de álgebra linear para maximizar a compressão da imagem mantendo um nível razoável de qualidade.

Modelando o Problema



$$\begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \\ 9 & 55 & 54 & 12 & 13 & 51 & 50 & 16 \\ 17 & 47 & 46 & 20 & 21 & 43 & 42 & 24 \\ 40 & 26 & 27 & 37 & 36 & 30 & 31 & 33 \\ 32 & 34 & 35 & 29 & 28 & 38 & 39 & 25 \\ 41 & 23 & 22 & 44 & 45 & 19 & 18 & 48 \\ 49 & 15 & 14 & 52 & 53 & 11 & 10 & 56 \\ 8 & 58 & 59 & 5 & 4 & 62 & 63 & 1 \end{bmatrix}$$

Figura : Subdivisão de uma imagem e sua matriz

Uma Transformação Linear

- Vejamos, num exemplo, como as wavelets de Haar agem em vetores (que serão as linhas da matriz da imagem).

Uma Transformação Linear

- Vejamos, num exemplo, como as wavelets de Haar agem em vetores (que serão as linhas da matriz da imagem).
- Se a matriz é de ordem 2^k a trasformação consistirá de k passos.

Uma Transformação Linear

- Vejamos, num exemplo, como as wavelets de Haar agem em vetores (que serão as linhas da matriz da imagem).
- Se a matriz é de ordem 2^k a trasformação consistirá de k passos.
- Quando temos uma linha da matriz de uma imagem 8×8, digamos

$$r = \begin{bmatrix} 420 & 680 & 448 & 708 & 1260 & 1420 & 1600 & 1600 \end{bmatrix}$$

teremos 3 passos.

Primeiro Passo

- Agrupe as coordenadas de r formando 4 pares

$$(420, 680), \quad (448, 708), \quad (1260, 1420) \quad \text{e} \quad (1600, 1600).$$

Primeiro Passo

- Agrupe as coordenadas de r formando 4 pares

$$(420, 680), \quad (448, 708), \quad (1260, 1420) \quad \text{e} \quad (1600, 1600).$$

- Determine as médias dos elementos em cada par

$$550, \quad 578, \quad 1340 \quad \text{e} \quad 1600.$$

Primeiro Passo

- Agrupe as coordenadas de r formando 4 pares

$$(420, 680), \quad (448, 708), \quad (1260, 1420) \quad \text{e} \quad (1600, 1600).$$

- Determine as médias dos elementos em cada par

$$550, \quad 578, \quad 1340 \quad \text{e} \quad 1600.$$

- Subtraia cada média da primeira coordenada do respectivo par

$$-130, \quad -130, \quad -80 \quad \text{e} \quad 0.$$

Primeiro Passo

- Agrupe as coordenadas de r formando 4 pares

$$(420, 680), \quad (448, 708), \quad (1260, 1420) \quad \text{e} \quad (1600, 1600).$$

- Determine as médias dos elementos em cada par

$$550, \quad 578, \quad 1340 \quad \text{e} \quad 1600.$$

- Subtraia cada média da primeira coordenada do respectivo par

$$-130, \quad -130, \quad -80 \quad \text{e} \quad 0.$$

- Construa um novo vetor

$$r_1 = \begin{bmatrix} 550 & 578 & 1340 & 1600 & -130 & -130 & -80 & 0 \end{bmatrix}.$$

Primeiro Passo

- Agrupe as coordenadas de r formando 4 pares

$$(420, 680), \quad (448, 708), \quad (1260, 1420) \quad \text{e} \quad (1600, 1600).$$

- Determine as médias dos elementos em cada par

$$550, \quad 578, \quad 1340 \quad \text{e} \quad 1600.$$

- Subtraia cada média da primeira coordenada do respectivo par

$$-130, \quad -130, \quad -80 \quad \text{e} \quad 0.$$

- Construa um novo vetor

$$r_1 = \begin{bmatrix} 550 & 578 & 1340 & 1600 & -130 & -130 & -80 & 0 \end{bmatrix}.$$

- Os quatro primeiros coeficientes de r_1 são chamados *coeficientes de aproximação* e os quatro últimos de *coeficientes de detalhe*.

Observação Importante

- O vetor r_1 é obtido através da multiplicação de r à direita por uma matriz W_1, a saber

$$W_1 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} \end{bmatrix}. \tag{1}$$

Segundo Passo

- Agrupe as 4 primeiras coordenadas do vetor r_1 formando 2 pares e calcule suas médias, como no primeiro passo.

Segundo Passo

- Agrupe as 4 primeiras coordenadas do vetor r_1 formando 2 pares e calcule suas médias, como no primeiro passo.
- Com isso obtemos as duas primeiras coordenadas de r_2, 564 e 1470, esses são os novos coeficientes de aproximação.

Segundo Passo

- Agrupe as 4 primeiras coordenadas do vetor r_1 formando 2 pares e calcule suas médias, como no primeiro passo.
- Com isso obtemos as duas primeiras coordenadas de r_2, 564 e 1470, esses são os novos coeficientes de aproximação.
- Subtraia cada média da primeira coordenada do respectivo par, essas são a terceira e quarta componentes de r_2: -14 e -130, os novos coeficientes de detalhe.

Segundo Passo

- Agrupe as 4 primeiras coordenadas do vetor r_1 formando 2 pares e calcule suas médias, como no primeiro passo.
- Com isso obtemos as duas primeiras coordenadas de r_2, 564 e 1470, esses são os novos coeficientes de aproximação.
- Subtraia cada média da primeira coordenada do respectivo par, essas são a terceira e quarta componentes de r_2: -14 e -130, os novos coeficientes de detalhe.
- As 4 últimas coordenadas de r_2 são as mesmas de r_1.

Segundo Passo

- Agrupe as 4 primeiras coordenadas do vetor r_1 formando 2 pares e calcule suas médias, como no primeiro passo.
- Com isso obtemos as duas primeiras coordenadas de r_2, 564 e 1470, esses são os novos coeficientes de aproximação.
- Subtraia cada média da primeira coordenada do respectivo par, essas são a terceira e quarta componentes de r_2: -14 e -130, os novos coeficientes de detalhe.
- As 4 últimas coordenadas de r_2 são as mesmas de r_1.
- Logo, o vetor r_2 é

$$r_2 = \begin{bmatrix} 564 & 1470 & -14 & -130 & -130 & -130 & -80 & 0 \end{bmatrix}.$$

Observação Importante

- O vetor r_2 é obtido através da multiplicação de r_1 à direita por uma matriz W_2, a saber

$$W_2 = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (2)$$

Terceiro Passo

- Calcule a média das duas primeiras coordenadas de r_2, este será o conteúdo da primeira posição de r_3.

Terceiro Passo

- Calcule a média das duas primeiras coordenadas de r_2, este será o conteúdo da primeira posição de r_3.
- Subtraia essa média da primeira coordenada de r_2, este será o conteúdo da segunda posição de r_3.

Terceiro Passo

- Calcule a média das duas primeiras coordenadas de r_2, este será o conteúdo da primeira posição de r_3.
- Subtraia essa média da primeira coordenada de r_2, este será o conteúdo da segunda posição de r_3.
- As 6 últimas coordenadas de r_3 são as mesmas de r_2.

Terceiro Passo

- Calcule a média das duas primeiras coordenadas de r_2, este será o conteúdo da primeira posição de r_3.
- Subtraia essa média da primeira coordenada de r_2, este será o conteúdo da segunda posição de r_3.
- As 6 últimas coordenadas de r_3 são as mesmas de r_2.
- Logo, o vetor r_3 é

$$r_3 = \begin{bmatrix} 1017 & -453 & -14 & -130 & -130 & -130 & -80 & 0 \end{bmatrix}.$$

Observação Importante

- Como antes, o vetor r_3 é obtido através da multiplicação de r_2 à direita por uma matriz W_3, a saber

$$W_3 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Usando W_1, W_2 e W_3

- Em vista das considerações a respeito das matrizes (1), (2) e (3) podemos obter r_3 diretamente de r observando que

$$r_3 = r_2 W_3 = r_1 W_2 W_3 = r W_1 W_2 W_3.$$

Usando W_1, W_2 e W_3

- Em vista das considerações a respeito das matrizes (1), (2) e (3) podemos obter r_3 diretamente de r observando que

$$r_3 = r_2 W_3 = r_1 W_2 W_3 = r W_1 W_2 W_3.$$

- Definindo $W = W_1 W_2 W_3$ temos

$$W = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & -\frac{1}{4} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & -\frac{1}{4} & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ \frac{1}{8} & -\frac{1}{8} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{8} & -\frac{1}{8} & 0 & \frac{1}{4} & 0 & 0 & -\frac{1}{2} & 0 \\ \frac{1}{8} & -\frac{1}{8} & 0 & -\frac{1}{4} & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{8} & -\frac{1}{8} & 0 & -\frac{1}{4} & 0 & 0 & 0 & -\frac{1}{2} \end{bmatrix} \tag{4}$$

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.
- $W = W_1 W_2 W_3$ também é ortogonal, logo, invertível.

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.
- $W = W_1 W_2 W_3$ também é ortogonal, logo, invertível.
- Como $r_3 = rW$ temos que $r = r_3 W^{-1}$.

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.
- $W = W_1 W_2 W_3$ também é ortogonal, logo, invertível.
- Como $r_3 = rW$ temos que $r = r_3 W^{-1}$.
- O que fizemos até agora foi operar somente com um vetor r, uma linha da matriz que representa imagem.

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.
- $W = W_1 W_2 W_3$ também é ortogonal, logo, invertível.
- Como $r_3 = rW$ temos que $r = r_3 W^{-1}$.
- O que fizemos até agora foi operar somente com um vetor r, uma linha da matriz que representa imagem.
- Para aplicar a wavelet em toda a imagem basta repetir esse processo a cada linha da matriz A.

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.
- $W = W_1 W_2 W_3$ também é ortogonal, logo, invertível.
- Como $r_3 = rW$ temos que $r = r_3 W^{-1}$.
- O que fizemos até agora foi operar somente com um vetor r, uma linha da matriz que representa imagem.
- Para aplicar a wavelet em toda a imagem basta repetir esse processo a cada linha da matriz A.
- Tipicamente aplica-se o processo novamente para cada coluna de A. Esse processo é conhecido como *transformada de Haar*.

"Teoria que ajuda na prática"

- As colunas das matrizes em (1), (2) e (3) são ortogonais, ou seja, elas formam bases ortogonais de $\mathbb{R}^8$.
- $W = W_1 W_2 W_3$ também é ortogonal, logo, invertível.
- Como $r_3 = rW$ temos que $r = r_3 W^{-1}$.
- O que fizemos até agora foi operar somente com um vetor r, uma linha da matriz que representa imagem.
- Para aplicar a wavelet em toda a imagem basta repetir esse processo a cada linha da matriz A.
- Tipicamente aplica-se o processo novamente para cada coluna de A. Esse processo é conhecido como *transformada de Haar*.
- Aplicar o processo em cada linha de A é multiplicá-la à direita por W e aplicar o processo em cada coluna de A é multiplicar AW à esquerda por W^T, obtendo

$$S = W^T AW \quad \text{e} \quad A = (W^T)^{-1} SW^{-1}.$$

Um exemplo

- Suponha que temos uma imagem cuja matriz é dada por

$$A = \begin{bmatrix} 640 & 1856 & 1856 & 1344 & 960 & 1856 & 1696 & 128 \\ 1088 & 1664 & 1408 & 800 & 544 & 288 & 1920 & 416 \\ 512 & 1216 & 576 & 832 & 704 & 960 & 1024 & 608 \\ 1696 & 1312 & 288 & 320 & 736 & 736 & 672 & 1504 \\ 1984 & 864 & 1888 & 1696 & 960 & 1888 & 544 & 1248 \\ 1472 & 1120 & 1728 & 1888 & 736 & 256 & 992 & 1504 \\ 1984 & 480 & 1888 & 864 & 512 & 1408 & 1344 & 1376 \\ 544 & 1568 & 640 & 416 & 416 & 1088 & 1408 & 1664 \end{bmatrix}.$$

- Então a matriz de A transformada é $S = W^{-1}AW$, ou seja,

$$S = \begin{bmatrix} 1102 & 99 & 49 & -125 & -10 & 132 & -182 & 72 \\ -97 & -16 & 111 & 51 & -254 & -28 & 70 & 260 \\ 149 & 95 & -180 & 10 & -184 & 176 & -48 & 436 \\ 99 & 167 & -158 & 120 & 124 & -152 & 140 & -116 \\ 138 & -46 & -156 & 312 & -160 & -24 & -288 & 16 \\ -52 & -8 & -260 & 92 & -272 & -56 & -64 & 312 \\ 86 & -58 & 36 & 320 & 192 & 88 & -352 & -48 \\ 132 & 124 & -168 & 96 & 632 & 200 & -56 & 56 \end{bmatrix}.$$

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.
- Matrizes com "grande proporção" de zeros são chamadas *esparsas* e gastam muito menos memória para serem armazenadas.

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.
- Matrizes com "grande proporção" de zeros são chamadas *esparsas* e gastam muito menos memória para serem armazenadas.
- O que fazer quando a matriz não é esparsa?

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.
- Matrizes com "grande proporção" de zeros são chamadas *esparsas* e gastam muito menos memória para serem armazenadas.
- O que fazer quando a matriz não é esparsa?
- Estabelecemos um valor positivo ε (um ponto de corte).

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.
- Matrizes com "grande proporção" de zeros são chamadas *esparsas* e gastam muito menos memória para serem armazenadas.
- O que fazer quando a matriz não é esparsa?
- Estabelecemos um valor positivo ε (um ponto de corte).
- Se $|s_{ij}| < \varepsilon$ então $s_{ij} := 0$. Senão, não altere s_{ij}.

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.
- Matrizes com "grande proporção" de zeros são chamadas *esparsas* e gastam muito menos memória para serem armazenadas.
- O que fazer quando a matriz não é esparsa?
- Estabelecemos um valor positivo ε (um ponto de corte).
- Se $|s_{ij}| < \varepsilon$ então $s_{ij} := 0$. Senão, não altere s_{ij}.
- Isso gera uma matriz um pouco mais esparsa, que gasta menos memória.

Como comprimir?

- Aplicando a transformada de Haar em regiões da imagem com pouca variação resulta numa matriz com muitos zeros.
- Matrizes com “grande proporção” de zeros são chamadas *esparsas* e gastam muito menos memória para serem armazenadas.
- O que fazer quando a matriz não é esparsa?
- Estabelecemos um valor positivo ε (um ponto de corte).
- Se $|s_{ij}| < \varepsilon$ então $s_{ij} := 0$. Senão, não altere s_{ij}.
- Isso gera uma matriz um pouco mais esparsa, que gasta menos memória.
- Se $\varepsilon = 0$ então não alteramos a matriz transformada e a imagem é recuperada sem perdas.

Como descomprimir?

- Ao iniciar o download de uma imagem o computador que a hospeda localiza a matriz codificada S e a envia ao solicitante.

Como descomprimir?

- Ao iniciar o download de uma imagem o computador que a hospeda localiza a matriz codificada S e a envia ao solicitante.
- Neste momento são enviados os coeficientes de aproximação e os maiores coeficientes de detalhe.

Como descomprimir?

- Ao iniciar o download de uma imagem o computador que a hospeda localiza a matriz codificada S e a envia ao solicitante.
- Neste momento são enviados os coeficientes de aproximação e os maiores coeficientes de detalhe.
- Em seguida são enviados os coeficientes de detalhe menores.

Como descomprimir?

- Ao iniciar o download de uma imagem o computador que a hospeda localiza a matriz codificada S e a envia ao solicitante.
- Neste momento são enviados os coeficientes de aproximação e os maiores coeficientes de detalhe.
- Em seguida são enviados os coeficientes de detalhe menores.
- Conforme o computador recebe os coeficientes de detalhe ele começa a reconstruir com cada vez mais precisão a imagem solicitada, até obtermos a imagem correspondente à matriz $(W^T)^{-1}SW^{-1}$.

Otimizações na compressão e descompressão

- Se utilizamos uma matriz W ortogonal ($W^{-1} = W^T$) então o processo de decodificação é muito mais rápido, pois

$$A = (W^T)^{-1} S W^{-1} = W S W^T,$$

ou seja, não precisamos calcular inversas.

Otimizações na compressão e descompressão

- Se utilizamos uma matriz W ortogonal ($W^{-1} = W^T$) então o processo de decodificação é muito mais rápido, pois

$$A = (W^T)^{-1} S W^{-1} = W S W^T,$$

  ou seja, não precisamos calcular inversas.
- Outra boa propriedade de matrizes ortogonais é que elas representam isometrias do espaço euclideano, e portanto preservam comprimento e ângulos.

Otimizações na compressão e descompressão

- Se utilizamos uma matriz W ortogonal ($W^{-1} = W^T$) então o processo de decodificação é muito mais rápido, pois

$$A = (W^T)^{-1} S W^{-1} = W S W^T,$$

ou seja, não precisamos calcular inversas.

- Outra boa propriedade de matrizes ortogonais é que elas representam isometrias do espaço euclideano, e portanto preservam comprimento e ângulos.
- Isto quer dizer menos distorções no processo de codificação da imagem, isto é, melhor qualidade.

Otimizações na compressão e descompressão

- Se utilizamos uma matriz W ortogonal ($W^{-1} = W^T$) então o processo de decodificação é muito mais rápido, pois

$$A = (W^T)^{-1} S W^{-1} = W S W^T,$$

ou seja, não precisamos calcular inversas.

- Outra boa propriedade de matrizes ortogonais é que elas representam isometrias do espaço euclideano, e portanto preservam comprimento e ângulos.
- Isto quer dizer menos distorções no processo de codificação da imagem, isto é, melhor qualidade.
- Para obter uma matriz ortogonal W basta observar que a matriz construída anteriormente tem vetores ortogonais nas colunas, bastando apenas dividir cada coluna pelo seu comprimento (como vetor do $\mathbb{R}^n$).

Como tratar um caso concreto?

- Seja $A = (a_{ij})$ a matriz que representa a imagem digital.

Como tratar um caso concreto?

- Seja $A = (a_{ij})$ a matriz que representa a imagem digital.
- O MATLAB interpreta cada entrada da matriz como um tom de cinza para o pixel correspondente de modo que o pixel com maior a_{ij} é branco puro e o com menor a_{ij} é preto puro.

Como tratar um caso concreto?

- Seja $A = (a_{ij})$ a matriz que representa a imagem digital.
- O MATLAB interpreta cada entrada da matriz como um tom de cinza para o pixel correspondente de modo que o pixel com maior a_{ij} é branco puro e o com menor a_{ij} é preto puro.
- Os outros pixels são preenchidos de acordo com uma escala entre estes dois valores.

Como tratar um caso concreto?

- Seja $A = (a_{ij})$ a matriz que representa a imagem digital.
- O MATLAB interpreta cada entrada da matriz como um tom de cinza para o pixel correspondente de modo que o pixel com maior a_{ij} é branco puro e o com menor a_{ij} é preto puro.
- Os outros pixels são preenchidos de acordo com uma escala entre estes dois valores.
- A função `image(A)` esboça uma figura que é representada pela matriz A, como descrito acima.

Nomes do dia-a-dia

- A matriz W que descrevemos acima é nada mais que uma matriz mudança de base da base canônica para uma outra base ortonormal de $\mathbb{R}^n$ na qual as coordenadas de uma dada linha da imagem tem coordenadas mais simples.

Nomes do dia-a-dia

- A matriz W que descrevemos acima é nada mais que uma matriz mudança de base da base canônica para uma outra base ortonormal de $\mathbb{R}^n$ na qual as coordenadas de uma dada linha da imagem tem coordenadas mais simples.
- "Taxa de compressão": Se escolhemos um ponto de corte $\varepsilon > 0$, estamos alterando a matriz S e portanto a imagem decodificada não será idêntica à original.

Nomes do dia-a-dia

- A matriz W que descrevemos acima é nada mais que uma matriz mudança de base da base canônica para uma outra base ortonormal de $\mathbb{R}^n$ na qual as coordenadas de uma dada linha da imagem tem coordenadas mais simples.
- "Taxa de compressão": Se escolhemos um ponto de corte $\varepsilon > 0$, estamos alterando a matriz S e portanto a imagem decodificada não será idêntica à original.
- O ponto é escolher o maior ε (matriz S mais esparsa) que não danifica seriamente a imagem.

Nomes do dia-a-dia

- A matriz W que descrevemos acima é nada mais que uma matriz mudança de base da base canônica para uma outra base ortonormal de $\mathbb{R}^n$ na qual as coordenadas de uma dada linha da imagem tem coordenadas mais simples.
- "Taxa de compressão": Se escolhemos um ponto de corte $\varepsilon > 0$, estamos alterando a matriz S e portanto a imagem decodificada não será idêntica à original.
- O ponto é escolher o maior ε (matriz S mais esparsa) que não danifica seriamente a imagem.
- A taxa de compressão é definida como a razão entre o número de entradas não nulas na matriz $S = W^T AW$ e o número de entradas não nulas na matriz obtida de S após a aplicação do "ponto de corte".