

Instituto Politécnico Nacional  
Escuela Superior de Cómputo

Compresión Wavelet

Procesamiento Digital de Señales

Integrantes:

Bautista Ríos Alfredo

Cisneros Araujo Karen

Contreras Vargas Oscar Daniel

Cortés Velazquez Samuel Alejandro

Ramírez Aguirre José Alfredo

Profesor:

Flores Escobar José Antonio

Este script de matlab comprime una imagen en escala de grises utilizando la transformada wavelet de Haar.

La compresión se lleva a cabo mediante la reducción de los coeficientes insignificantes en la representación wavelet de la imagen.

### Código

```
1. %Archivo:          wavecomp.m
2. %Equipo:           5
3. %Intergantes:      Bautista Ríos Alfredo
4. %                  Cisneros Araujo Karen
5. %                  Contreras Vargas Oscar Daniel
6. %                  Cortés Velazquez Samuel Alejandro
7. %                  Ramírez Aguirre José Alfredo
8. %practica :Compresion Wavelet
9. %Compresion de imagenes por medio de una Wavelet
10.%La Wavelte a utilizar es la Haar
11.clear all;
12.close all;
13.clc;
14.
15.%Lectura de la imagen original
16.imagen = double(imread("tif.png"));
17.%figure(1)
18.%imshow(imagen);
19.
20.%delta = 0.0001;
21.%delta = 0.05;
22.delta = 0.005;

23.%Definir la Wavelet Haar
24.%Mediante 3 matrices: H1, H2,H3
25.
26.H1=[0.5 0 0 0 0.5 0 0 0;
27.0.5 0 0 0 -0.5 0 0 0;
28.0 0.5 0 0 0 0.5 0 0 ;
29.0 0.5 0 0 0 -0.5 0 0 ;
30.0 0 0.5 0 0 0 0.5 0;
31.0 0 0.5 0 0 0 -0.5 0;
32.0 0 0 0.5 0 0 0 0.5;
33.0 0 0 0.5 0 0 0 -0.5];
34.H2=[0.5 0 0.5 0 0 0 0 0;
35.0.5 0 -0.5 0 0 0 0 0;
36.0 0.5 0 0.5 0 0 0 0;
37.0 0.5 0 -0.5 0 0 0 0;
38.0 0 0 0 1 0 0 0;
39.0 0 0 0 0 1 0 0;
40.0 0 0 0 0 0 1 0;
41.0 0 0 0 0 0 0 1];
42.H3=[0.5 0.5 0 0 0 0 0 0;
43.0.5 -0.5 0 0 0 0 0 0;
44.0 0 1 0 0 0 0 0;
45.0 0 0 1 0 0 0 0;
46.0 0 0 0 1 0 0 0;
```

```

47. 0 0 0 0 0 1 0 0;
48. 0 0 0 0 0 0 1 0;
49. 0 0 0 0 0 0 0 1;];
50.
51. %Normalizar cada una de las columnas de las 3 matrices
52. %Con esto se asegura que los resultados de la normalizacion
53. %son ortonormales para cada columna de la matriz
54. H1o = (H1.*(2^0.5));
55. H2o = (H2.*(2^0.5));
56. H3o = (H3.*(2^0.5));
57.
58. %Multiplicar las 3 matrices
59. H = H1o * H2o * H3o;
60. %Esta seria la matriz resultante, normalizada
61. Ho = norm(H);
62. %Multiplicar las tres matrices de Haar
63. H = H1 * H2 * H3;
64.
65. %Tamaño de la imagen
66. len = length(size(imagen));
67.
68. %El ejemplo no aplica para una imagen RGB
69.
70. if len~=2
71. error("Se necesita una imagen con escala de grises");
72. %Si se desea aplicar este ejemplo a una imagen RGB, usar
73. %un ejemplo con wavelet rgb(haar_wt_rgb)
74. End
75.
76. %Crear dos imagenes con valores 0
77. yo = zeros(size(imagen));
78. y = yo;
79. %Se obtienen los renglones y las columnas de la imagen original
80. [r,c] = size(imagen);
81.
82. %A continuacion se realiza la transformada de la matriz H
83. %En este caso se va a operar en bloques de 8x8
84. for i=0 : 8 : r-8
85. for j=0 : 8 : c-8
86. p=i+1;
87. q=j+1;
88. yo (p:p+7,q:q+7) = Ho' * imagen(p:p+7,q:q+7) * Ho;
89. y (p:p+7,q:q+7) = H' * imagen(p:p+7,q:q+7) * H;
90. end
91. end
92. %Mostrar la imagen original
93. figure(1)
94. imshow(imagen/255);
95. title('Imagen original');
96. %Determinar el numero de valores no zeros de y
97. n1 = nnz(y);
98.
99. %crea una nueva imagen
100. zo=yo;
101. %Aqui obtivimos el valor maximo que esta entre 0 y 1

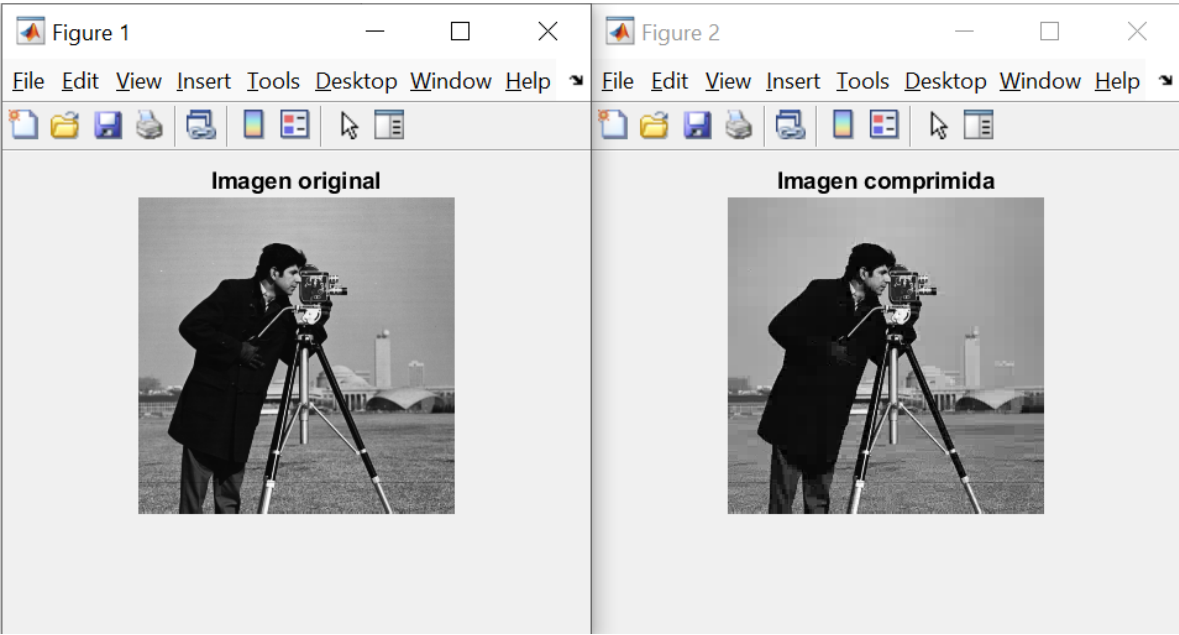
```

```



102. m = max(max(yo));
103.
104. yo = yo / m;
105. %Los valores que esten entre +delta y -delta en Y, se reemplazan por
ceros
106. %esto es un parametro que ayuda en la compresion
107. yo(abs(yo)<delta) = 0;
108. yo = yo*m;
109.
110. %crea una nueva imagen
111. z=y;
112. y = y / m;
113. %Los valores que esten entre +delta y -delta en Y, se reemplazan por
ceros
114. %esto es un parametro que ayuda en la compresion
115. y(abs(y)<delta) = 0;
116. y = y*m;
117.
118. %Determinar el numero de calores no ceros de y
119. n2 = nnz(y);
120.
121. %Segunda parte de la practica
122. for i=0 : 8 : r-8
123.     for j=0 : 8 : c-8
124.         p=i+1;
125.         q=j+1;
126.         zo (p:p+7,q:q+7) = Ho * yo(p:p+7,q:q+7) * Ho';
127.         z(p:p+7,q:q+7) = inv(H') * y(p:p+7,q:q+7) * inv(H);
128.     end
129. end
130. figure(2)
131. imshow(z/255);
132. title('Imagen comprimida');
133.
134. %Por ultimo, obtener un factor de compresion aproximado
135. factorcomp = n2/n1;
136.
137. %Guardar las imagenes
138. imwrite(imagen/255,'original.tif');
139. imwrite(z/255,'comprimida.tif');

```

Ejecución



Este proceso reduce la cantidad de datos necesarios para representar la imagen, eliminando información que se considera menos significativa según el umbral delta, por eso no logramos ver a simple vista un cambio significativo entre ambas imágenes, sin embargo, con notar el peso de ambas imágenes logramos ver que la compresión fue exitosa.

 imagen_comprimida	20/06/2024 05:07 p. m.	Archivo PNG	20 KB
 imagen_original	20/06/2024 05:20 p. m.	Archivo PNG	38 KB