Procesamiento Digital de Imágenes

Pablo Roncagliolo B.
Nº 16

Orden de las clases...

- CAPTURA, DIGITALIZACION Y ADQUISICION DE IMAGENES ✓
- TRATAMIENTO ESPACIAL DE IMAGENES ✓
- TRATAMIENTO EN FRECUENCIA DE IMAGENES ✓
- RESTAURACION DE IMAGENES ✓
- OPERACIONES MORFOLOGICAS ✓
- PROCESAMIENTO DE IMÁGENES EN COLORES ✓
- COMPRESION DE IMAGENES ←
- REPRESENTACION Y DESCRIPCION
- SEGMENTACION DE IMAGENES
- TOPICOS AVANZADOS

Compresión de Imágenes

A continuación se profundizará en algunos aspectos de compresión de imágenes tratados en el libro de Gonzalez & Wood

Compresión de Imágenes

Sea n1 y n2 el número de datos de dos conjuntos que representan la misma información. Si n1<>n2 entonces existe redundancia de datos

Entonces se define R_D como la "redundancia de datos" y C_R como la "razón de compresión".

$$R_D = 1 - \frac{1}{C_R}$$

$$C_R = \frac{n_1}{n_2}$$

Ej. Si n_1=10*n_2 entonces:

C_R = 10 = 10:1

R_D= 0.9 ➔90% datos en n1 son redundantes

$$n_2 = n_1 \rightarrow C_R = 1 \rightarrow R_D = 0 \quad \text{No hay redundancia}$$
$$n_2 << n_1 \rightarrow C_R = \infty \rightarrow R_D = 1 \quad \text{Alta redundancia}$$

Compresión de Imágenes

Redundancia en la codificación de la información:

Sea M una imagen que posee resolución en frecuencia de sólo 8 niveles de gris: $[r_0, r_1..r_7]=[0, 1/7, ..7/7]$

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

TABLE 8.1 Example of variable-length coding.

Luego de analizar el histograma de la imagen se determina la probabilidad de ocurrencia del nivel r_k como: $p_r(r_k)=n_k/n$

Donde "n_k" es el número de ocurrencia del valor "r_k" y "n" es el número total de píxeles

Compresión de Imágenes

La codificación estándar sería "Code 1" y el número de bits requeridos para cada nivel se indica en "$l_r(r_k)$"

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

TABLE 8.1 Example of variable-length coding.

El promedio ponderado de bits requerido para la codificación es:

$$L_{avg} = \sum_{k=0}^{7} l(r_k)\,p_r(r_k)$$

Para "Code1" el prom. de bits es $L_{avg1}=3$ y para "Code 2" es $L_{avg2}=2.7$

Entonces la razón de compresión es $C_R=3/2.7=1.11$

La redundancia de datos es $R_D=1-1/1.11=0.099$ ➔ 9.9% datos redundantes en "Code1"

Compresión de Imágenes

El esencia de la compresión por código de largo variable es que se asigna códigos “cortos” a niveles de gris con mayor ocurrencia...

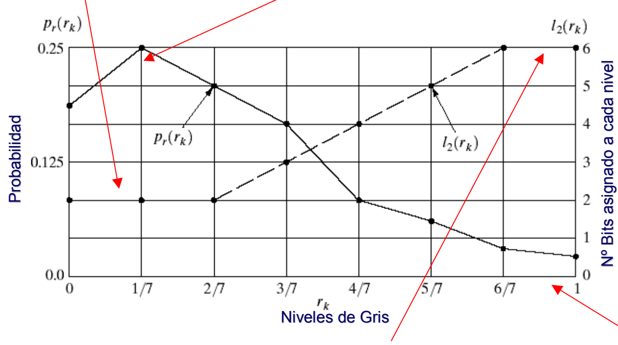
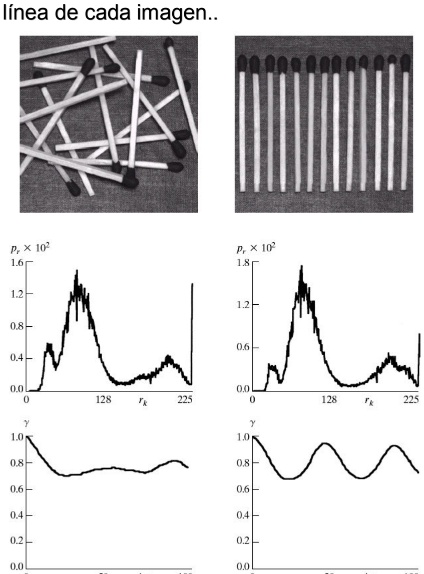


FIGURE 8.1 Graphic representation of the fundamental basis of data compression through variable-length coding.

... y códigos “largos” a niveles de gris con menor ocurrencia

Compresión de Imágenes

Redundancia entre píxeles: ambas imágenes poseen un histograma similar.
Sin embargo calculando un coeficiente denominado de “autocorrelación” sobre una línea de cada imagen..



a b
c d
e f

FIGURE 8.2 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

... se observa que para la imagen “b” existe una mayor correlación cada 45 y 90 píxeles.

Esto corresponde a la separación aproximada entre los objetos.

Existe redundancia entre píxeles ➔ algún método de compresión puede aprovechar esta situación!

Compresión de Imágenes

Redundancia visual: La imagen “a” tiene 256 niveles de gris. La imagen “b” posee sólo 16 niveles de gris ➔ razón de compresión 2:1, pues de 8 bits por píxel se reduce a 4 bits por píxel. Sin embargo se generan “falsos bordes”.

a b c

FIGURE 8.4
(a) Original image.
(b) Uniform quantization to 16 levels. (c) IGS quantization to 16 levels.

Adicionando “ruido” al último bit en la imagen “c” se elimina dicha percepción manteniendo la razón de compresión 2:1

Compresión de Imágenes

Criterios de Fidelidad en la compresión: Se utilizan criterios absolutos como el E_{rms} (error cuadrático medio) y la SNR (relación señal a ruido) entre la imagen original y la comprimida.

$E_{rms\ b}$=6.93
$E_{rms\ c}$=6.78

SNR_b=10.25
SNR_c=10.39

...mmm..no dice mucho...

También se utilizan criterios subjetivos (tablas) como:

TABLE 8.3
Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Imagen “b”=marginal

Imagen “c”=passable

Compresión de Imágenes

Modelo general para compresión y transmisión

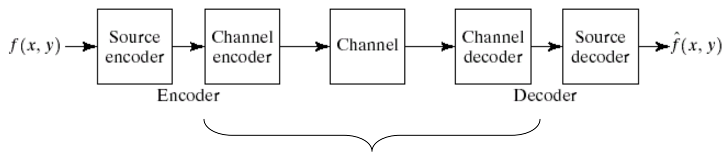


FIGURE 8.5 A general compression system model.

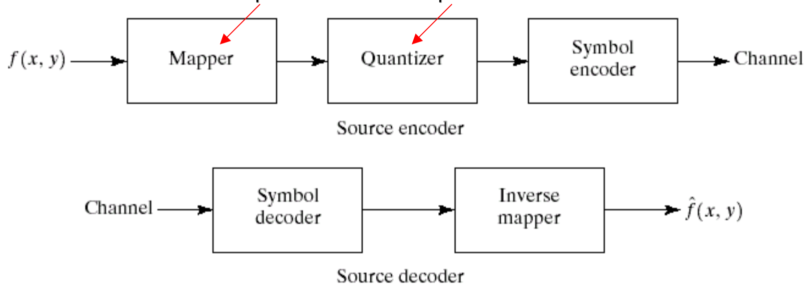
Si no hay transmisión se omite estos pasos

Compresión de Imágenes

Modelo general para la codificación

sin pérdida

con pérdida



a
b

FIGURE 8.6 (a) Source encoder and (b) source decoder model.

Compresión de Imágenes

Elementos de Teoría de la Información

¿Cuál es la cantidad mínima de datos para representar una imagen sin pérdida de información?

Sea E un evento que ocurre con probabilidad P(E) entonces la información "I" es:

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

Es decir a mayor probabilidad P(E) menor información I(E)

Por ejemplo si el evento "E" ocurre con probabilidad "1" la información es "0". O sea no hay "ninguna novedad".

Por ejemplo si el evento "E" ocurre con probabilidad "0.99" la información es muy baja. Por el contrario si se transmite el evento "not E" de probabilidad "0.01" la información será muy alta!

La base del logaritmo corresponde a la unidad utilizada para medir la información. Si la base es "2" la unidad de información se mide en "bit".

Compresión de Imágenes

Elementos de Teoría de la Información

Dada una fuente de información que puede emitir sólo los símbolos "a_j" pertenecientes a "A" cada uno con probabilidad "$P(a_j)$" que conforman el vector "z", entonces la fuentes de información queda representada por el conjunto (A,z):

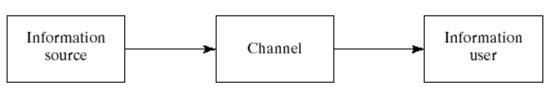


FIGURE 8.7 A simple information system.

Ensemble $(A, \mathbf{z})$
$A = \{a_j\}$
$\mathbf{z} = [P(a_1), P(a_2), \ldots, P(a_J)]^T$

$Q = [q_{kj}]$

Ensemble $(B, \mathbf{v})$
$B = \{b_k\}$
$\mathbf{v} = [P(b_1), P(b_2), \ldots, P(b_K)]^T$

La información que se produce al emitir el símbolo "a_j" es:

$$I(a_j) = -\log P(a_j)$$

Compresión de Imágenes

Elementos de Teoría de la Información

La información que se produce al emitir el símbolo "a_j" es:

$$I(a_j) = -\log P(a_j)$$

Si se emiten k símbolos del conjunto "A", la cantidad individual de símbolos emitidos será:

$$kP(a_1) + kP(a_2) + \ldots + kP(a_j)$$

Por lo tanto la cantidad total de información transmitida será:

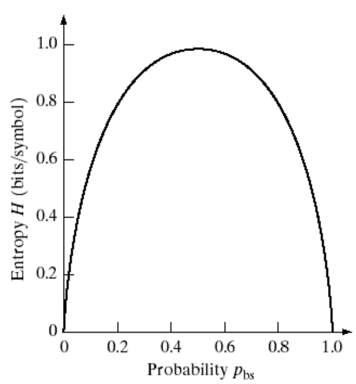
$$-kP(a_1)\log P(a_1) - kP(a_2)\log P(a_2) \ldots - kP(a_j)\log P(a_j)$$

Entonces el promedio de información (sumando y dividiendo por k) es:

$$H(z) = -\sum_{j=1}^{J} P(a_j)\log P(a_j)$$

Esta cantidad H(z) es la entropía de la fuente, que corresponde al promedio de información de la fuente. Mayor entropía (incertidumbre, desorden) mayor información!

Compresión de Imágenes



$$H(z) = -\sum_{j=1}^{J} P(a_j)\log P(a_j)$$

Entropía de la información en un ensayo de Bernoulli X (experimento aleatorio en que X puede tomar los valores 0 o 1). La entropía depende de la probabilidad P(X=1) de que X tome el valor 1. Cuando P(X=1)=0.5, todos los resultados posibles son igualmente probables, por lo que el resultado es poco predecible y la entropía es máxima.

Ej.

$$H(z) = -\sum_{j=1}^{2} 0.5\log_2 0.5 = -\sum_{j=1}^{2} 0.5(-1) = 1$$

Ej.

$$H(z) = -0.1\cdot\log_2 0.1 - 0.9\cdot\log_2 0.9 = 0.468$$

Compresión de Imágenes

Codificación Huffman

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

FIGURE 8.11 Huffman source reductions.

FIGURE 8.12 Huffman code assignment procedure.

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Compresión de Imágenes

Source symbol	Probability	Binary Code	Huffman	Truncated Huffman	B_2-Code	Binary Shift	Huffman Shift
Block 1							
a_1	0.2	00000	10	11	C00	000	10
a_2	0.1	00001	110	011	C01	001	11
a_3	0.1	00010	111	0000	C10	010	110
a_4	0.06	00011	0101	0101	C11	011	100
a_5	0.05	00100	00000	00010	C00C00	100	101
a_6	0.05	00101	00001	00011	C00C01	101	1110
a_7	0.05	00110	00010	00100	C00C10	110	1111
Block 2							
a_8	0.04	00111	00011	00101	C00C11	111 000	00 10
a_9	0.04	01000	00110	00110	C01C00	111 001	00 11
a_{10}	0.04	01001	00111	00111	C01C01	111 010	00 110
a_{11}	0.04	01010	00100	01000	C01C10	111 011	00 100
a_{12}	0.03	01011	01001	01001	C01C11	111 100	00 101
a_{13}	0.03	01100	01110	10 0000	C10C00	111 101	00 1110
a_{14}	0.03	01101	01111	10 0001	C10C01	111 110	00 1111
Block 3							
a_{15}	0.03	01110	01100	10 0010	C10C10	111 111 000	00 00 10
a_{16}	0.02	01111	010000	10 0011	C10C11	111 111 001	00 00 11
a_{17}	0.02	10000	010001	10 0100	C11C00	111 111 010	00 00 110
a_{18}	0.02	10001	001010	10 0101	C11C01	111 111 011	00 00 100
a_{19}	0.02	10010	001011	10 0110	C11C10	111 111 100	00 00 101
a_{20}	0.02	10011	011010	10 0111	C11C11	111 111 101	00 00 1110
a_{21}	0.01	10100	011011	10 1000	C00C00C00	111 111 110	00 00 1111
Entropy	4.0						
Average length		5.0	4.05	4.24	4.65	4.59	4.13

TABLE 8.5 Variable-length codes.

Huffman se acerca a la codificación óptima. Sin embargo posee alta complejidad computacional. Huffman truncado genera códigos sólo para símbolos más probables. (desde el 13 aplica código de prefijo)

MATLAB Command Window

File Edit View Window Help

```
>> p=[0.2 0.1 0.1 0.06 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.02 0.02 0.02 0.02 0.02 0.01];
>> Entropia=sum(-p.*log2(p))

Entropia =

    4.0015
```

Compresión de Imágenes

Planos de Bit:

Corresponde a la separación de cada píxel de la imagen en los 8 bits del byte *(para el caso de imágenes de 8bits)*

Esto genera 8 planos: bit_0, bit_1,... Bit_7

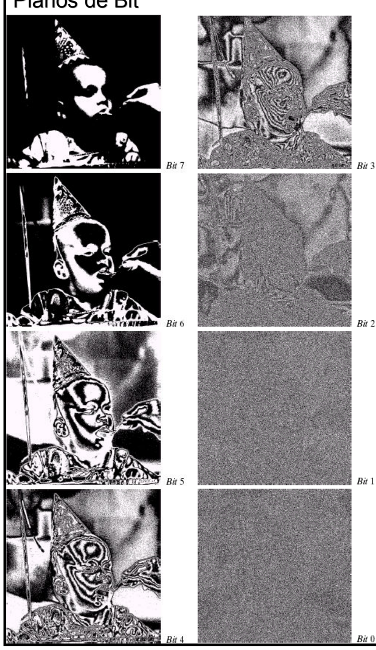
El OR o la SUMA de todos los planos corresponde a la imagen original

Bit 7

```
I=imread('foto.jpg');
G=double(rgb2gray(I));
% Bit más significativo
plano=7;
% bitand es un AND bit a bit en Matlab
Plano1=bitand(G,2^plano)/2^plano;
imshow(Plano1)
```


Compresión de Imágenes

Planos de Bit

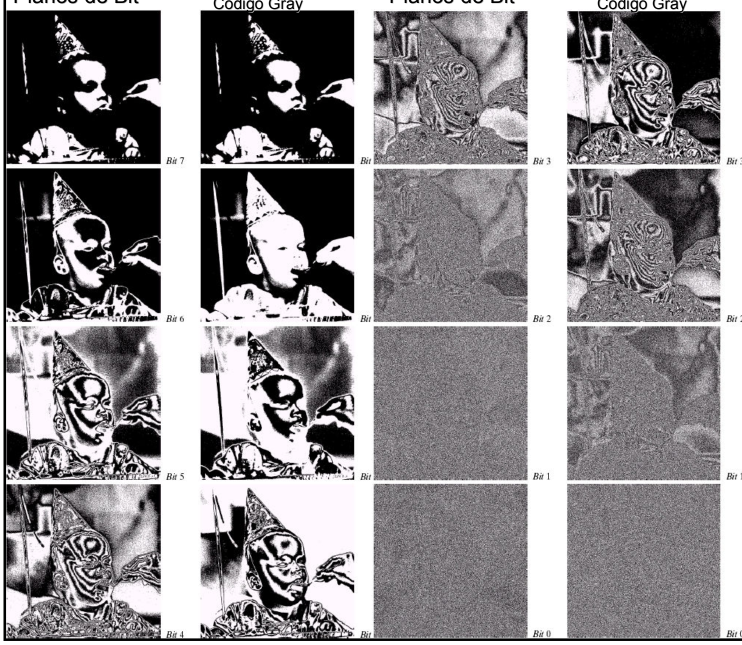



```
% Ej. Para visualizar cada plana
I=imread('foto.jpg');
G=double(rgb2gray(I));
For plano=0:7
 Plano1=bitand(G,2^plano)/2^plano;
 imshow(Plano1,[]);pause(1);
end;
```

```
% Ej. Para reconstruir progresivamente
I=imread('foto.jpg');
G=double(rgb2gray(I));
plano=7;
Plano1=bitand(G,2^plano);
imshow(Plano1),pause(1)
plano=6;
Plano1=Plano1+bitand(G,2^plano);
imshow(Plano1,[]),pause(1)
plano=5;
...
...
```

Compresión de Imágenes

Planos de Bit | Planos de Bit: Código Gray | Planos de Bit | Planos de Bit: Código Gray



Los planos con codificación Gray son menos “complejos”. Tienes zonas con colores similares ➔ posibilita mejor compresión

Ej.
127=01111111
128=10000000

Gray:
127=11000000
128=01000000

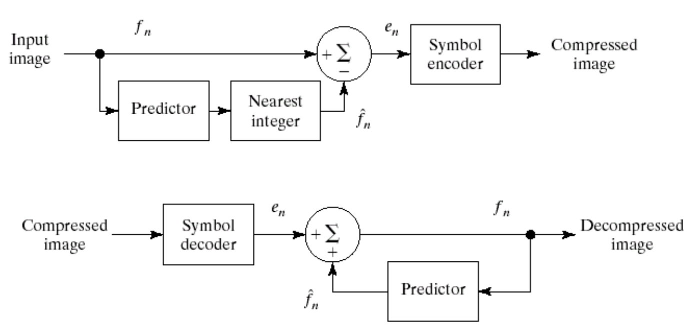
Compresión de Imágenes

Codificación predictiva sin pérdida:

Se transmite sólo la nueva información que aporta cada píxel.

a
b

FIGURE 8.19 A lossless predictive coding model: (a) encoder; (b) decoder.



Por ejemplo el “predictor” puede ser la moda de los últimos 3 píxeles.
Ej. 128, 128,129 ➔ precitor ➔128 ➔ nueva información “1”