

## 6 Compresión de imágenes

En nuestros días, una enorme cantidad de información se maneja por medios digitales. La compresión es necesaria en aplicaciones de almacenamiento, transmisión, tiempo real. En este capítulo veremos los fundamentos de la compresión en la teoría de la información, la terminología utilizada en el campo de la compresión, y algunos modelos que se aplican comúnmente a las imágenes.

### 6.1 Fundamentos

#### Compresión y redundancia de datos

Se requiere distinguir entre los términos *datos* e *información*. Para nosotros, los datos son el medio por el que se transporta la información. Esto quiere decir que se pueden utilizar distintas cantidades de datos para representar la misma cantidad de información.

El término *compresión de datos* se refiere al proceso de reducir la cantidad de datos requeridos para representar una cantidad dada de información.

Cuando un conjunto de datos contiene más del mínimo necesario para transmitir la información, decimos que existe *redundancia de datos*.

La redundancia de datos se puede representar matemáticamente de la siguiente forma: Si  $n_1$  y  $n_2$  denotan el número de unidades que cargan información (por ejemplo, bits) en 2 conjuntos de datos representando la misma información, la *redundancia relativa de datos*  $R_D$  del primer conjunto de datos se define como:

$$R_D = 1 - 1/C_R \quad (6.1)$$

Donde  $C_R$  representa la *relación de compresión* y es

$$C_R = n_1/n_2 \quad (6.2)$$

Por ejemplo, si  $n_2 = n_1$ ,  $C_R = 1$  y  $R_D = 0$ , quiere decir que la primera representación no contiene datos redundantes (relativo a la segunda).

Cuando  $n_2 \ll n_1$ ,  $C_R \rightarrow \infty$  y  $R_D \rightarrow 1$ , quiere decir que hay una compresión considerable y datos altamente redundantes.

Cuando  $n_2 \gg n_1$ ,  $C_R \rightarrow 0$  y  $R_D \rightarrow -\infty$ , quiere decir que el segundo conjunto contiene mucho más datos que la representación original (el indeseable caso de *expansión de datos*).

$C_R$  y  $R_D$  están generalmente en los intervalos abiertos  $(0, \infty)$  y  $(-\infty, 1)$ , respectivamente.

Una relación de compresión de 10 (o 10:1) significa que el primer conjunto tiene 10 unidades por cada unidad del segundo conjunto. La redundancia correspondiente de 0.9 implica que 90% de los datos en el primer conjunto es redundante.

En el caso de la compresión de imágenes, se identifican y explotan 3 tipos de redundancia de datos: *redundancia de código*, *redundancia entre píxeles* y *redundancia psicovisual*. Al reducir o eliminar una o más de estas conseguimos comprimir los datos.

### 6.1.1 Redundancia de código

Utilizaremos el ya familiar concepto de histograma de niveles de gris de una imagen para mostrar cómo éste puede ayudar en gran medida en la construcción de códigos que reducen la cantidad de datos utilizados para representar la imagen.

Se asume que la variable aleatoria discreta  $r_k$  en el intervalo  $[0, 1]$  representa los niveles de gris de una imagen y que cada  $r_k$  ocurre con una probabilidad  $p_r(r_k)$  :

$$p_r(r_k) = n_k/n \quad k = 0, 1, 2, \dots, L - 1 \quad (6.3)$$

donde  $L$  es el número de niveles de gris,  $n_k$  es el número de veces que el nivel  $k$  aparece en la imagen, y  $n$  es el número total de píxeles en la imagen.

Si el número de bits utilizado para representar cada valor de  $r_k$  es  $l(r_k)$ , entonces el número promedio de bits utilizados para representar cada pixel será

$$L_{prom} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k) \quad (6.4)$$

Es decir que la longitud promedio de las palabras código asignadas a los valores de los niveles de gris se halla sumando el producto del número de bits empleados para representar cada nivel de gris y la probabilidad de que aparezca ese nivel. Así el número total de bits requeridos para codificar una imagen  $M \times N$  es  $MNL_{prom}$ .

Si se representan los niveles de gris con un código binario natural de  $m$  bits,

$$L_{prom} = m \quad (6.5)$$

| $r_k$       | $p_r(r_k)$ | 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          |

Por ejemplo, en la tabla se muestra la distribución de niveles de gris de una imagen de 8 niveles. Si se utiliza un código natural de 3 bits (Code 1) para representar los 8 niveles,  $L_{avg} = 3$  porque  $l_1(r_k) = 3$  para todo  $r_k$ . Si se utiliza el código 2, el número promedio de bits es

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

$$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08) + 5(0.06) + 6(0.03) + 5(0.02)$$

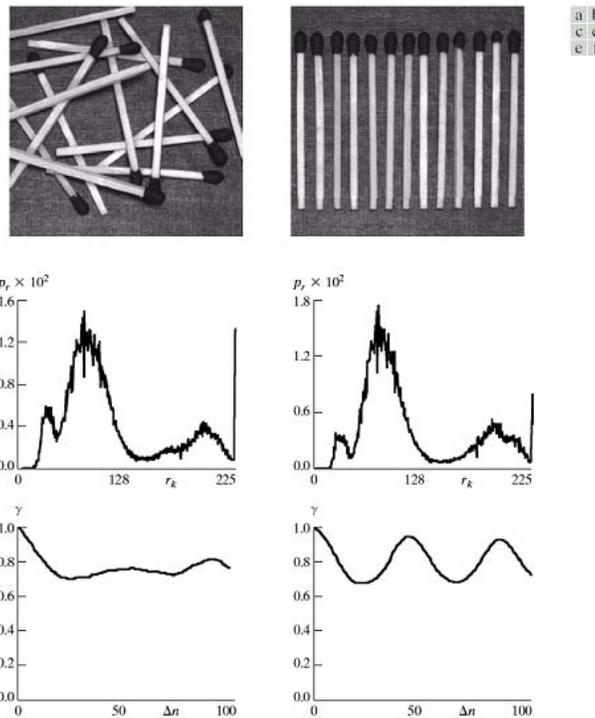
$$= 2.7 \text{ bits}$$

En el ejemplo anterior, asignar menos bits a los niveles de gris más probables y más bits a los menos probables se tradujo en una compresión de los datos. A este proceso se le denomina *codificación de longitud variable*.

La redundancia de código aparece cuando los códigos asignados a un conjunto de sucesos (cada nivel de gris) no han sido seleccionados de manera que se obtenga el mayor rendimiento posible de las probabilidades de que ocurran estos sucesos.

### 6.1.2 Redundancia entre píxeles

Consideremos las imágenes de la figura 6.2. Sus histogramas son prácticamente idénticos, con 3 rangos de nivel de gris dominantes. Se puede usar un código del tipo que acabamos de ver para reducir la redundancia de datos. Sin embargo, este proceso de codificación no alteraría el nivel de correlación entre los píxeles de las imágenes.



6.1.2 imágenes, sus histogramas y sus diagramas de correlación

Es decir que los códigos empleados para representar los niveles de gris de cada imagen no tienen nada que ver con la correlación<sup>1</sup> entre píxeles, estas correlaciones resultan de relaciones estructurales o geométricas entre los objetos de la imagen.

<sup>1</sup> La correlación no se verá en este curso. Para profundizar, consulte Gonzalez & Woods (2002) o Jahne (2003).

Las figuras e y f muestran los coeficientes de autocorrelación calculados a lo largo de una línea de cada imagen. Estos perfiles se pueden relacionar cualitativamente con la estructura de las imágenes de las imágenes correspondientes, especialmente en la segunda figura donde la elevada correlación de los pixeles separados se puede relacionar directamente con el espacio entre cada cerillo.

Además, los pixeles adyacentes de ambas imágenes están muy correlacionados.

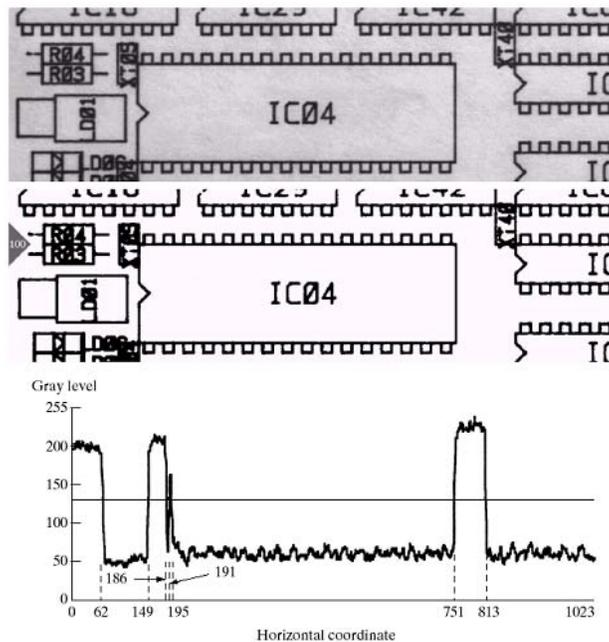
Estas ilustraciones reflejan otra forma de redundancia de los datos. *Ya que es posible predecir el valor de un pixel a partir de sus vecinos, la información que aporta individualmente un pixel es relativamente pequeña.*

Es decir, la mayor parte de la contribución visual de un pixel se puede inferir de los valores de sus vecinos (es redundante).

Para reducir estas redundancias entre pixeles, la distribución bidimensional se transforma a un formato más eficaz, usualmente no visualizable.

Por ejemplo, se podría utilizar las diferencias entre pixeles adyacentes para representar una imagen.

A este tipo de transformaciones se les conoce como *correspondencias o mapeos*. Se dice que son *reversibles* si se pueden reconstruir los elementos originales a partir del conjunto de datos transformados.



Line 100: (1,63) (0,87) (1,37) (0,5) (1,4) (0,556) (1,62) (0,210)

## 6.2 Un mapa de bits, su imagen binaria, el diagrama de niveles de gris de la línea 100 y su codificación por longitud de códigos variable

En la figura se muestra un procedimiento de mapeo sencillo. La figura muestra un diagrama electrónico muestreado a 330 dpi, y su versión binaria. Ya que la imagen binaria contiene muchas regiones de intensidad constante, una representación más

eficiente puede ser construída mapeando los píxeles en cada línea de rastreo  $f(x,0)$ ,  $f(x,y)$ , ...,  $f(x, N-1)$  en una secuencia de pares  $(g_1, w_1)$ ,  $(g_2, w_2)$ , ..., donde  $g_i$  denota el  $i$ -ésimo nivel de gris en la línea y  $w_i$  el código de longitud de la  $i$ -ésima corrida. En la figura se muestra el código de longitud variable de la línea 100. Sólo 88 bits se necesitan para representar los 1024 bits de la línea.

### 6.1.3 Redundancia psicovisual

En el capítulo 2 vimos cómo la iluminación de una región tal como la percibe el ojo humano depende de muchos factores. El ojo no responde con la misma sensibilidad a toda información visual. Cierta información tiene menos importancia relativa que otra en nuestro proceso visual normal. Se dice que esta información es psicovisualmente redundante y se puede eliminar sin que se altere significativamente la calidad de la percepción de la imagen.

En general un observador busca características diferenciadoras, tales como bordes o regiones de diferentes texturas, y las combina mentalmente en grupos reconocibles. El cerebro relaciona estos grupos con el fin de completar la interpretación de la imagen.

Al contrario que las redundancias ya vistas, la redundancia psicovisual está asociada a la información visual real o cuantificable. Su eliminación es posible porque la propia información no es esencial para el procesamiento visual normal.

Como la eliminación de este tipo de datos se traduce en una pérdida de información cuantitativa, a menudo se denomina con el nombre de *cuantificación*.

Ya que es una operación irreversible, la cuantificación conduce a una compresión con pérdida de datos.



6.3 Imagen original de 8 bits, su cuantificación a 4 bitsy la cuantificación mejorada a 4 bits

La figura muestra la imagen original con 256 niveles de gris y una imagen cuantificada a 16 niveles de gris. Como lo vimos en el capítulo 2, la cuantificación conduce a la presencia de falso contorno. La tercera imagen muestra la imagen cuantificada a 16 niveles de gris con un método que toma en cuenta las peculiaridades del sistema visual

humano, llamado *cuantificación de escala de grises mejorada* (IGS). Consiste en añadir a cada pixel un componente pseudoaleatorio con el fin de eliminar los cambios bruscos tales como los del falso contorno.

La tabla muestra el método. Se toman los 4 bits menos significativos del nivel de gris pixel anterior y se le suman al actual. Los 4 bits más significativos de la suma forman el código IGS. Si los 4 bits más significativos del valor actual son 1111, se suma 0000.

| Pixel   | Gray Level | Sum       | IGS Code |
|---------|------------|-----------|----------|
| $i - 1$ | N/A        | 0000 0000 | N/A      |
| $i$     | 0110 1100  | 0110 1100 | 0110     |
| $i + 1$ | 1000 1011  | 1001 0111 | 1001     |
| $i + 2$ | 1000 0111  | 1000 1110 | 1000     |
| $i + 3$ | 1111 0100  | 1111 0100 | 1111     |

#### 6.1.4 Criterios de fidelidad

El quitar datos psicovisualmente redundantes tiene como resultado una pérdida de información real o cuantitativa. Existen 2 clases de criterios utilizados para cuantificar la pérdida de información: El criterio de fidelidad objetivo y el criterio de fidelidad subjetivo.

Cuando el nivel de pérdida de información puede expresarse como una función de la imagen original y la imagen comprimida (y posteriormente descomprimida), se dice que se basa en un *criterio de fidelidad objetiva*.

Uno de estos criterios es el error raíz media cuadrada ( $e_{rms}$ ) entre una imagen de entrada y una de salida.

Si  $f(x,y)$  es la imagen de entrada y  $g(x,y)$  es la aproximación de  $f(x,y)$  resultante de comprimir y después descomprimir la entrada, para cada valor de  $x$  y  $y$ , el *error* entre las dos se define como

$$e(x,y) = g(x,y) - f(x,y) \quad (6.6)$$

y el *error total* entre las dos imágenes es

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [g(x,y) - f(x,y)] \quad (6.7)$$

Donde las imágenes son de tamaño  $M \times N$ . El  $e_{rms}$  entre  $f(x,y)$  y  $g(x,y)$  es la raíz cuadrada del error al cuadrado en promedio en el arreglo  $M \times N$

$$e_{rms} = [1/MN \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [g(x,y) - f(x,y)]^2]^{1/2} \quad (6.8)$$

Otro criterio de fidelidad relacionado es la relación media cuadrada señal - ruido de la imagen comprimida-descomprimida. Si  $g(x,y)$  se considera como la suma de la imagen original  $f(x,y)$  y la señal de ruido  $e(x,y)$ , la relación media cuadrada señal - ruido de la imagen de salida es

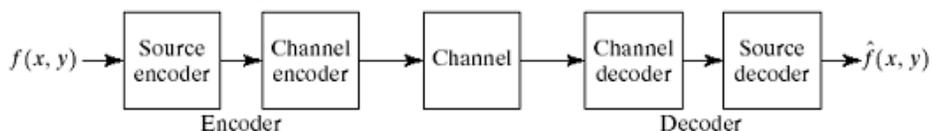
$$SNR_{rms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [g(x,y) - f(x,y)]^2} \quad (6.9)$$

El valor rms de la razón señal-ruido, denotado  $SNR_{rms}$  se obtiene de la raíz cuadrada de  $SNR_{ms}$ .

Los criterios de fidelidad subjetiva se aplican mostrando una imagen a un grupo de observadores y promediando sus evaluaciones. Usualmente se utilizan valores como *mucho peor, peor, ligeramente peor, igual, ligeramente mejor, mejor, mucho mejor* para evaluar.

## 6.2 Modelos de compresión de imágenes

Las técnicas de compresión de datos vistas en la sección anterior usualmente son combinadas para formar sistemas prácticos de compresión de imágenes. En esta sección examinaremos las características generales de un sistema de este tipo y desarrollaremos un modelo general para representarlo.



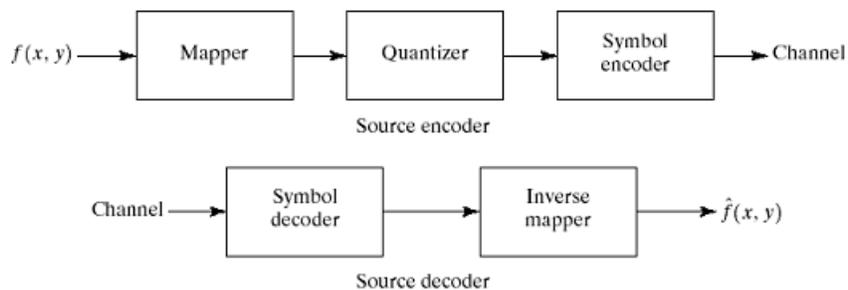
6.4 Modelo general de sistema de compresión

Una imagen  $f(x,y)$  se introduce en un *codificador*, el cuál crea un conjunto de símbolos para representarla. Este conjunto se transmite por medio de un *canal* a un *decodificador*, donde se reconstruye una imagen  $g(x,y)$ .  $g(x,y)$  puede o no ser una réplica exacta de  $f(x,y)$ , si lo es, se dice que el sistema es libre de error o que preserva la información, si no, hay algún nivel de distorsión en la imagen de salida.

El codificador está formado por un *codificador de fuente*, que remueve las redundancias de la entrada, y un *codificador de canal*, que aumenta la inmunidad al ruido de la salida del codificador de fuente. El *decodificador de canal* y *decodificador de fuente* que componen el decodificador realizan las funciones inversas. Si el canal está libre de ruido, se puede prescindir del codificador y decodificador de canal.

### 6.2.1 El codificador y el decodificador de fuente

Normalmente, se modela un codificador de fuente con una serie de 3 operaciones independientes diseñadas para reducir una de las 3 redundancias vistas anteriormente.



6.5 Codificador y decodificador de fuente

En la primera parte, el *mapeador* transforma los datos de entrada a un formato diseñado para reducir la redundancia entre píxeles. Esta operación es usualmente reversible y puede o no resultar en compresión de datos. La codificación por reducción de longitud de código es un ejemplo de mapeado que resulta en compresión.

La segunda parte, o el *cuantificador*, reduce la precisión de la salida del mapeador de acuerdo a un criterio de fidelidad preestablecido (esta etapa reduce la redundancia psicovisual). Esta operación es irreversible (debe ser omitida si se quiere compresión sin errores).

En la etapa final el *codificador de símbolos* crea un código de longitud variable que representa la salida del cuantificador y mapea la salida de acuerdo al código, asignando las palabras clave más cortas a los valores más frecuentes, reduciendo la redundancia de código.

Esta operación es reversible.

El decodificador de fuente de la figura contiene sólo 2 componentes: el *decodificador de símbolos* y el *mapeador inverso*. Estos bloques realizan en orden invertido las operaciones inversas del codificador de símbolos de la fuente y el mapeador.

### 6.2.2 El codificador y decodificador de canal

Estos están diseñados para reducir el impacto de ruido en el canal insertando redundancia controlada en los datos codificados por la fuente.

Una de las técnicas más utilizadas de codificación de canal es el código de Hamming (R.W. Hamming [1950]), basado en añadir suficientes bits a los datos codificados para asegurarse de que un número mínimo de bits debe cambiar entre palabras de código válidas.

Hamming mostró que si 3 bits de redundancia se añaden a una palabra de 4 bits, de modo que la *distancia*<sup>2</sup> entre 2 palabras válidas es 3, se puede detectar y corregir cualquier error *de un solo bit*.

La palabra de código Hamming de 7 bits (7, 4)  $h_1h_2\dots h_5h_6h_7$  asociada a un número binario de 4 bits  $b_4b_3b_2b_1$  es

$$\begin{array}{ll} h_1 = b_3 \oplus b_2 \oplus b_0 & h_3 = b_3 \\ h_2 = b_3 \oplus b_1 \oplus b_0 & h_5 = b_2 \\ h_4 = b_2 \oplus b_1 \oplus b_0 & h_6 = b_1 \\ & h_7 = b_0 \end{array}$$

donde + en un círculo denota la operación OR exclusiva.

Nótese que los bits  $h_1$ ,  $h_2$  y  $h_4$  son bits de paridad par<sup>3</sup> para los campos de bits  $b_3b_2b_0$ ,  $b_3b_1b_0$  y  $b_2b_1b_0$ , respectivamente.

---

<sup>2</sup> La distancia entre 2 palabras de código se define como el número mínimo de dígitos que debe cambiar en una palabra para obtener la otra palabra. Por ejemplo, la distancia entre 101101 y 100111 es 2.

Para decodificar, se revisa que si valor codificado tenga paridad impar en los campos de bits en que se estableció paridad par. Si la palabra de paridad  $c_4c_2c_1$  es diferente de cero, hay un error

$$\begin{aligned}c_1 &= h_1 \oplus h_3 \oplus h_5 \oplus h_7 \\c_2 &= h_2 \oplus h_3 \oplus h_6 \oplus h_7 \\c_4 &= h_4 \oplus h_5 \oplus h_6 \oplus h_7\end{aligned}$$

y el decodificador simplemente complementa el bit en la posición indicada por la palabra de paridad. Después se extrae el valor binario decodificado de la palabra corregida como  $h_3h_5h_6h_7$

---

<sup>3</sup> Una cadena de bits binarios tiene paridad par si el número de bits con valor 1 es par.