Series de Fourier: Métodos de compresión mediante las DCT

Aguilera Alexis

Estudiante de Ingeniería en Sistemas de Computación
Universidad Nacional del Sur, Avda. Alem 1253, B8000CPB Bahía Blanca, Argentina
alex_aguilera9@hotmail.com
Junio 2014

Resumen: el objetivo de este trabajo es mostar la implementación de compresión de señales de imágenes mediante el uso de DCT(Transformada Discreta del Coseno). Para esto se debe considerar que la energía de vídeo de la imagen tiene una frecuencia espacial bastante baja, que varía lentamente con el tiempo. Por tanto una transformada puede concentrar la energía en muy pocos coeficientes.

Palabras clave: Series de Fourier, Compresión de imágenes, Algoritmo de compresión con pérdida JPEG

I. INTRODUCCIÓN

Tanto a la hora de almacenar como de transmitir una señal, resulta trascendental una correcta codificación de la misma con el fin de llevar a cabo una operación eficiente. En este sentido, muy importante resulta la compresión de los datos, de tal forma que se encuentre un compromiso entre las dos características deseables que son la no pérdida de información relevante y un pequeño volumen con el fin de que se minimicen los recursos dedicados a su almacenaje o transmisión.

Existen varios métodos de compresión de datos que utilizan las DCT, en este texto abordaremos solamente el algoritmo de compresión con pérdida JPEG (del ingles Joint Photographic Experts Group, Grupo Conjunto de Expertos en Fotografía).

II. TRANSFORMADA DISCRETA DEL COSENO (DCT)

Nuestro propósito a la hora de emplear una transformada ortogonal para representar una imagen en un nuevo dominio es el de estar en condiciones de lograr un alto grado de compresión de datos sin sacrificar apenas calidad visual.

La DCT nos permite calcular las componentes de proyección de la señal en el dominio original sobre el espacio de transformación compuesto por las funciones base tipo coseno. En otras palabras, la señal se puede expresar como combinación lineal de un conjunto de funciones que definen una base en el espacio de transformación. Debido a que la DCT es una transformada ortogonal, los coeficientes de transformación (las funciones base tipo coseno, las cuales se pueden apreciar en la figura 1) cumplen con las propiedades necesarias de ortonormalidad y completitud. La propiedad de completitud asegura que la transformada es invertible y la de ortonormalidad garantiza que la energía se localiza en los primeros coeficientes de la transformación, lo que equivale a decorrelar las componentes del vector de señal de entrada en mayor o menor grado, cosa que es mucho más eficiente en el dominio transformado.

La DCT bidimensional aplicable a imágenes puede llevarse a cabo computando dos transformadas monodimensionales gracias a que la transformada es separable. Por tanto, la ecuación de análisis o transformada directa se define como:

Donde:

Por otro lado, la ecuación de síntesis o transformación inversa se define de la forma:

Como ya hemos expuesto, la DCT concentra mucho la información en los primeros coeficientes, por lo que podemos aplicar técnicas para reducir la cantidad de datos en el dominio transformado sin perder apenas información relevante que nos lleve a una pérdida notable de calidad de la imagen en el dominio espacial.

III. ALGORITMO DE COMPRESIÓN CON PÉRDIDA (JPEG)

Algoritmo de compresión con pérdida se refiere a cualquier procedimiento de codificación que tenga como objetivo representar cierta cantidad de información utilizando una menor cantidad de la misma, siendo imposible una reconstrucción exacta de los datos originales.

En el primer caso, en lugar de guardar dos copias inexactas, sólo se guarda una aproximación. Esta aproximación se aprovecha de las limitaciones del sistema visual humano para esconder la distorsión introducida. Estos algoritmos son de gran utilidad para guardar imágenes fotográficas que de otra manera ocuparían mucho espacio dificultando su transmisión y almacenamiento. Un ejemplo de algoritmo con pérdida de calidad es JPEG.

Este algoritmo cuenta de los siguientes pasos:

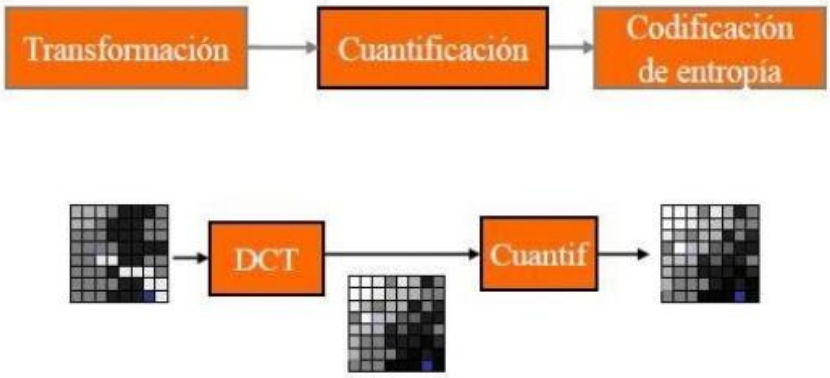

Figura 1. Algoritmo de Compresión JPEG

A. Preparación de la Entrada inicial

La imagen original es dividida en matrices de 8 por 8. Los colores son convertidos al sistema YUV; utilizando este sistema es más fácil eliminar información innecesaria de la imagen ya que separa la intensidad de los colores de su tonalidad. Opcionalmente puede promediarse la información de píxeles vecinos en los canales U y V para reducir la cantidad de información a ser codificada.

B. Transformación DCT

En cada matriz de 8 por 8 se debe calcular la DCT bidimensional. La DCT permite expresar cada matriz en el dominio de las frecuencias (este proceso no comprime la información, solo la expresa de otra forma). Cada uno de los cuadrados es una combinación lineal de los 64 restantes.

Un ejemplo de uno de esos pequeños bloques de 8×8 inicial es este:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

El siguiente proceso es restarles 128 para que queden números entorno al 0, entre -128 y 127.

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Se procede a la transformación por DCT de la matriz, y el redondeo de cada elemento al número entero más cercano.

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

Nótese que el elemento más grande de toda la matriz aparece en la esquina superior izquierda; este es el coeficiente DC.

C. Cuantificación Digital

Como el ojo humano no es bueno detectando cambios de brillo en pequeñas áreas (variación de alta frecuencia) se pueden eliminar estas altas frecuencias, en este proceso es donde se pierde la mayor información(calidad). El resultado de esto es que los componentes de las altas frecuencias, tienden a igualarse a cero, mientras que muchos de los demás, se convierten en números positivos y negativos pequeños.

Una matriz de cuantificación típica es la matriz de Losheller que se usa opcionalmente en el estándar JPEG:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Dividiendo cada coeficiente de la matriz de la imagen transformada entre cada coeficiente de la matriz de cuantificación, se obtiene esta matriz, ya cuantificada:

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Por ejemplo, cuantificando el primer elemento, el coeficiente DC, sería así:

$$\mathrm{round}\left(\frac{-415}{16}\right) = \mathrm{round}(-25.9375) = -26$$

D. Cuantificación entrópica

La codificación entrópica es una forma especial de la compresión sin pérdida de datos. Para ello se toman los elementos de la matriz siguiendo una forma de zig-zag, poniendo grupos con frecuencias similares juntos, e insertando ceros de codificación, y usando la codificación Huffman para lo que queda.

En la matriz anterior, la secuencia en zig-zag, es esta:
−26, −3, 0, −3, −2, −6, 2, −4, 1 −4, 1, 1, 5, 1, 2, −1, 1, −1, 2, 0, 0, 0, 0, 0, −1, −1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

JPEG tiene un código Huffman para cortar la cadena anterior en el punto en el que el resto de coeficientes sean ceros, y así, ahorrar espacio:
−26, −3, 0, −3, −2, −6, 2, −4, 1 −4, 1, 1, 5, 1, 2, −1, 1, −1, 2, 0, 0, 0, 0, 0, −1, −1, EOB

E. Decoficación

Es el mismo proceso pero de manera inversa.
-Se decodifica y se pone el valor en su casilla.
-Se multiplica el valor por el correspondiente a la matriz de cuantificación usada.
-Se deshace la transformación DCT.
-Se suman 128 a cada entrada.

IV. CONCLUSIÓN

A modo de conclusión podemos decir que a la hora de comprimir una imagen, con los conocimientos de Transformadas de Fourier vistos en clase y algún lenguaje de programación podemos implementar este algoritmo sencillamente.

Esto nos da una pauta de que los temas vistos en la materia Funciones de Variable Compleja tienen una aplicación casi directa en problemas reales

REFERENCIAS

[1] Wikipedia, La enciclopedia libre, [internet] disponible en http://es.wikipedia.org/wiki/Joint_Photographic_Experts_Group y http://es.wikipedia.org/wiki/Algoritmo_de_compresi%C3%B3n_con_p%C3%A9rdida
[2]Algoritmo JPEG [Internet] disponible en http://es.slideshare.net/csacanam/algoritmo-jpeg
[3] Comparación entre la Aplicación de la DCT y la KLT a la Compresión de Imágenes Digitales [Internet] disponible en http://www.ilopez.es/proyectos/teoriadelasenal/CompImagKLT.pdf
[4] Wavelet Tour on Signal Processing” (Mallat S. A.) Academy Press 1999.