

Guión de prácticas nº 6:

El estándar de compresión JBIG

Rubén Dugo Martín
Compresión y Codificación de Datos
Curso 2007-2008
Universidad de Granada

1. **Comprimir, usando los parámetros por defecto, la imagen ptt1.pbm. Comprobar cómo es posible descomprimirla a diferentes resoluciones usando, por ejemplo, -y 600. Se puede usar el programa display para visualizar las salidas.**

Me he sorprendido bastante cuando he comprimido el fichero, ya que se ha alcanzado la razón de compresión de 30.50:1, esto está íntimamente relacionando con el contenido del fichero, puesto que la mayoría de él son píxeles en blanco.

He comprobado con distintas imágenes binarias y no se llegan a comprimir tanto.

Por otro al lado, al descomprimir la imagen con distintas resoluciones se observa claramente la disminución de tamaño de la salida.

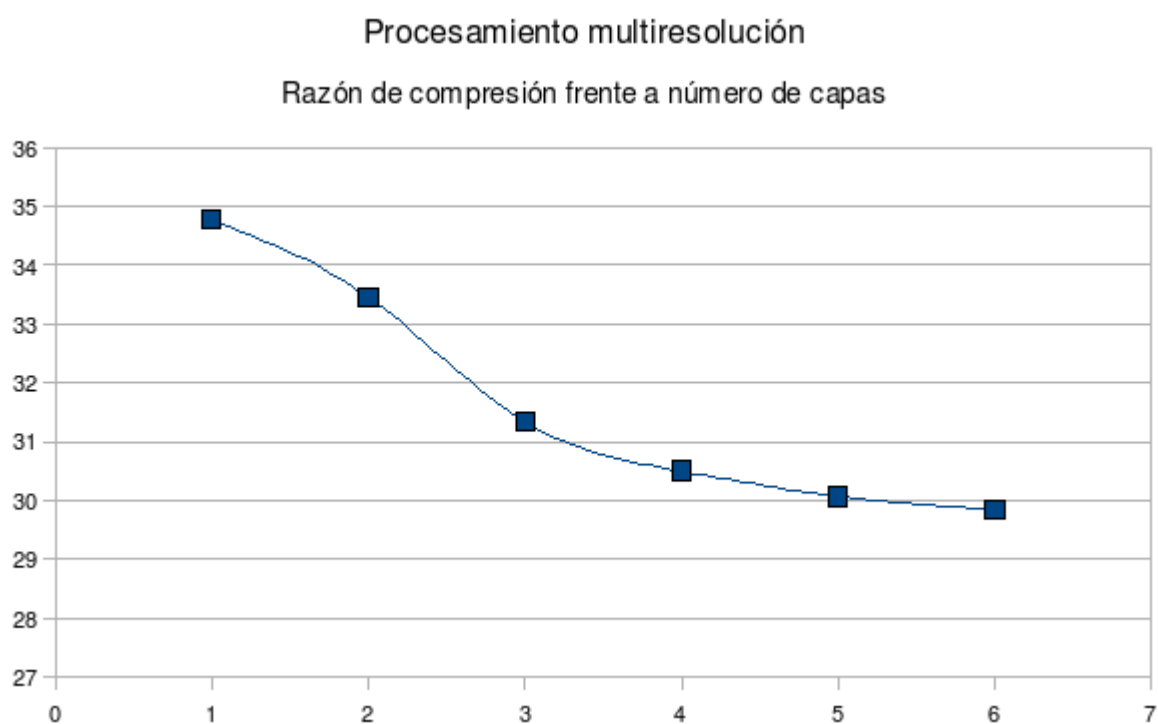
2. **Comprobar como incluir la funcionalidad de procesamiento multiresolución, permitiendo diferentes capas, tiene un costo en el número de bits del fichero comprimido.**

Para ésto es necesario usar el parámetro *-d*, aquí muestro una tabla con las razones de compresión obtenidas con distintos números de capas de diferencia.

Nota: Con el número de capas de diferencia igual a cero se anula el procesamiento multiresolución.

RAZÓN DE COMPRESIÓN OBTENIDA CON DISTINTOS NÚMEROS DE CAPAS DE CAPAS DE DIFERENCIA.	
Nº de capas de diferencia	Ratio
0	34.77:1
1	33.46:1
2	31.34:1
3	30.50:1
4	30.07:1
5	29.85:1

Para que se más ilustrativo, lo podemos representar gráficamente:



Como se puede observar tanto en la tabla como en la gráfica, la razón de compresión disminuye conforme aumenta el número de capas, podemos afirmar sin lugar a dudas que *el procesamiento multiresolución tiene un costo en el número de bits del fichero comprimido* .

3. Codificar una imagen en escala de grises usando código Gray y binario para los planos de bits. Comparar el tamaño de los ficheros resultantes.

Probando con distintas imágenes llegamos a la conclusión de que los ficheros ocupan exactamente lo mismo, ya que codificar una cadena de bits con código Gray no altera su longitud.

4. Comprimir varias imágenes binarias y en escala de grises y comparar los resultados con los obtenidos usando codificación aritmética (usando el programa sample de Michael Dipperstein).

Usando dichos programas con los parámetros por defecto, obtenemos la siguiente información:

RAZONES DE COMPRESIÓN DE DIFERENTES IMÁGENES MEDIANTE CODIFICACIÓN ARITMÉTICA Y JBIG.		
<i>Imágenes binarias (pbm)</i>		
Fichero	Ratio (jbig)	Ratio (aritmética)
ppt1.pbm	30.50:1	11.27:1
ppt4.pbm	8.74:1	3.92:1
ppt8.pbm	33.91:1	4.95:1
<i>Imágenes en escala de grises (pgm)</i>		
Fichero	Ratio (jbig)	Ratio (aritmética)
camera.pgm	1.66:1	1.13:1
bird.pgm	1.94:1	1.17:1
bridge.pgm	1.23:1	1.03:1

En la tabla se ve claramente que en todos los casos la compresión con JBIG es mayor, en algunos casos en concreto la diferencia es enorme, se puede observar que las razones de compresión obtenidas con JBIG varían razonablemente de unas imágenes a otras, esto está relacionado con el contenido de los ficheros.

Podemos afirmar que la codificación JBIG es mucho más óptima que la aritmética, al menos para estos ficheros.

5. Opcionalmente, se sugiere crear un programa básico para comprimir/descomprimir imágenes pbm/pgm a jbig usando jbiglib y las rutinas de entrada/salida del alumno. En caso de imágenes pgm, el alumno puede preprocesar la imagen para que se calcule la diferencia de cada píxel de la imagen con el anterior o con un entorno (anterior) de cada pixel. El decodificador debe postprocesar la imagen resultante para recuperar la información. Nótese que se necesitan 9 bits para poder codificar las diferencias en lugar de los 8 que tiene cada pixel de una imagen en escala de grises.

Para este ejercicio el realizado el programa *jbigbasic*, el cual incluyo en el paquete *jbigkit.tar.gz*.

Para compilarlo simplemente hay que ejecutar *make* en el directorio del paquete, una vez compilado se puede ejecutar con los siguientes parámetros:

Uso: jbigbasic <accion> [opciones] <fichero entrada> <fichero salida>

El campo accion debe ser uno de los siguientes:

- c codificar*
- d decodificar*

Y las opciones:

-p preprocesar/postprocesar la imagen calculando la diferencia de un pixel con el anterior (solo para PGM).

Es muy estricto leyendo los parámetros así que hay que introducirlos en el orden que describe, sólo soporta imágenes PGM o PBM.

El código está bastante comentado y debido a la falta de tiempo no he podido darle más funcionalidad (aunque me hubiera gustado).

El postprocesamiento de una imagen que ha sido preprocesada calculando la diferencia de sus píxeles está sin acabar, lo he intentado implementar utilizando la función *jbg_dec_getimage(...)* de la biblioteca JBIG que me devolvería un plano que se correspondería al signo de la diferencia y otros planos para los demás bits de los píxeles, de esta forma tendría menos trabajo.