



APENDICE A. PRUEBAS DEL JJ2000

A.1 Software que incluye JPEG2000

En estos momentos no hay muchas aplicaciones que hagan uso de JPEG2000 [9], aunque está previsto que las empresas de software empiecen a hacer uso de este nuevo estándar de compresión en todo tipo de aplicaciones. Los usos para los que está previsto este formato gráfico son mucho más avanzados que el resto de estándares de compresión de imágenes tales como JPEG. Entre las principales herramientas que ya están implementadas para hacer uso de JPEG2000, podemos nombrar las siguientes:

- JJ2000: Implementación en Java de la parte 1 de JPEG2000. Actualmente en versión de desarrollo 5.1. Incluye un Apple para visualizar imágenes J2K. Forma parte del software de referencia del estándar JPEG2000 y es de libre distribución [13].
- Jasper: Es una colección de software (librerías y aplicaciones) para la codificación y manipulación de imágenes. Entre otros formatos gráficos es capaz de manejar JPEG2000 JP2 y JPEG2000 codestream. Está programado en C y contiene *30.000 líneas de código*. Forma parte del software de referencia del estándar JPEG2000, también es de libre distribución [10-12].
- Kakadu: Es un conjunto de herramientas fuertemente optimizada para desarrolladores de JPEG2000. Implementa en C++ todos los aspectos de la parte 1 del estándar.
- Migrator 2000: Tiene como objetivo desarrollar, demostrar y diseminar herramientas para la norma de imágenes digitales JPEG2000. El proyecto comprende el archivado y distribución de imágenes así como su utilización en aplicaciones médicas, incluyendo la



implementación de protección de los derechos de autor (IPR), descripción de contenido y marcas de agua.

A.2 Pruebas del JJ2000

Quizá la mayoría de los códigos libres que proporcionan algunos organismos de investigación suelen venir con errores de compilación, por ello se ha hecho este apéndice para la respectiva prueba del Sistema JJ2000 con el fin de verificar su correcto funcionamiento.

Se han encontrado diferentes versiones del JJ2000; está la versión 4.1 y versión 5.1. con las que se pueden trabajar, además de estas versiones, también hay una parte de Interfaz Gráfica de Usuario con la que se puede controlar mediante ventanas y cuadros de diálogo los diferentes parámetros que se le proporcionan al JJ2000 para su funcionamiento.

Las diferentes versiones del JJ2000 se pueden descargar de la página oficial [13]. El software JJ2000 está conformado por tres paquetes

- El código fuente.
- El código en lenguaje Java
- Una completa documentación en HTML del JJ2000.

La primera versión del JJ2000, versión 4.1, proporciona gran cantidad de ventajas, contiene un codificador así como un decodificador en el que se recogen las principales características del estándar JPEG2000. Las funcionalidades apoyadas por este *codec* están sujetas con el algoritmo de la parte 1 de JPEG2000 tal cual como está definido en el Estándar Final Preliminar Internacional del JPEG2000.



A.2.1. Descompresión de los archivos

Se crea un directorio donde se va a instalar el código JJ2000. Dentro de este directorio se descomprime el archivo *jj2000_4.1.zip*, el cual genera un directorio “*jj2000-4.1*” con todos los archivos necesarios para el funcionamiento del JJ2000. Los archivos que se encuentran en el directorio “*jj2000-4.1*” una vez descomprimido son los siguientes:

- *jj2000-4.1.jar*
- README
- COPYRIGHT

El funcionamiento de los archivos depende de la plataforma que se vaya a utilizar, para este caso se usa la Máquina Virtual de JAVA (JVM) además del Sun's JDK.

Antes de comenzar se debe asignar el CLASSPATH necesario para el funcionamiento del respectivo JDK, dependiendo del tipo de sistema operativo que se utilice varía la forma de ejecución del CLASSPATH. A continuación se muestran las instrucciones que se deben tener en cuenta para su funcionamiento. Para este proyecto se trabajará bajo el sistema operativo *Windows (XP)*.

- Windows 95/98/NT/XP:

```
set CLASSPATH=<jjdir>/jj2000-4.1.jar; %CLASSPATH %
```

- UNIX con Shell de Bourne:

```
CLASSPATH=<jjdir>/jj2000-4.1.jar:$CLASSPATH export CLASSPATH
```

- UNIX con C-Shell:

```
setenv CLASSPATH <jjdir>/jj2000-4.1.jar:$CLASSPATH
```

o

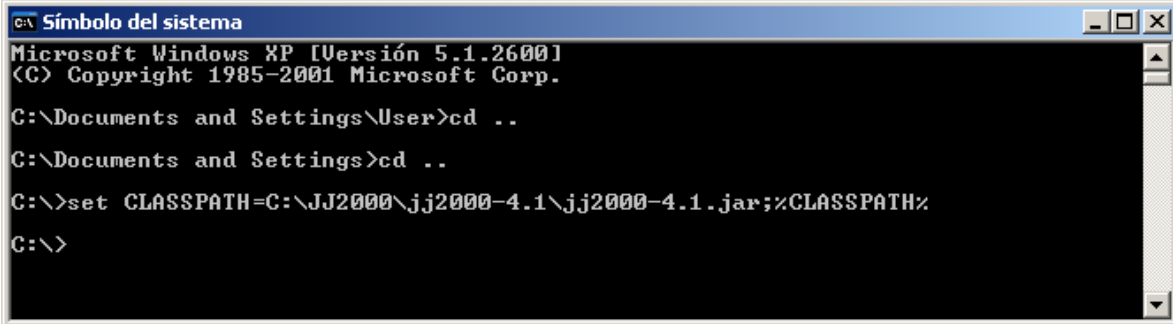
```
setenv CLASSPATH <jjdir>/jj2000-4.1.jar
```



por si el CLASSPATH ya ha sido definido.

Se debe tener en cuenta que el <jjdir> es el directorio donde se descomprime el archivo 'jj2000-4.1.jar'. En este trabajo se ha creado un <jjdir> en C:\JJ2000\jj2000-4.1. Para acceder a este directorio se debe de escribir en MS-DOS o Símbolo del sistema las siguientes líneas de comando:

```
set CLASSPATH=C:\JJ2000\JJ2000-4.1\jj2000-4.1.jar;%CLASSPATH%
```



```
Símbolo del sistema
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\User>cd ..
C:\Documents and Settings>cd ..
C:\>set CLASSPATH=C:\JJ2000\jj2000-4.1\jj2000-4.1.jar;%CLASSPATH%
C:\>
```

Con estos comandos en ejecución queda encendido el JJ2000 listo para ser configurado.

A.2.2. Ejecución del Codificador y Decodificador

Una vez asignado el CLASSPATH se procede a escribir dentro del MS-DOS las respectivas líneas de código necesarias para el control del CODEC.

Para controlar el codificador se tiene que escribir "java JJ2KEncoder <args>" y para el decodificador: "java JJ2KDecoder <args>", donde "<args>" es el argumento que especifica los parámetros que se necesitan para el funcionamiento del CODEC, por ejemplo el argumento "-u" imprime un mensaje del terminal en uso mostrando varios detalles como el sistema operativo utilizado, cantidad de memoria RAM en uso, etc. El argumento "-i" hace referencia al archivo de entrada al decodificador el cual puede ser un archivo en formato .ppm, .pgm, que son las extensiones de los archivos que se van a comprimir; mientras el argumento "-o" direcciona el sitio donde se guardará el archivo comprimido, obteniendo en la salida un archivo con extensión .j2k; el argumento '-v' imprime un



mensaje en pantalla de la versión y los derechos de autor de JJ2000. Para esta prueba utilizamos la imagen de Lena mostrada en la *figura 1*, la cual tiene un tamaño de 128KB.



Figura 1. Lena.pgm

Al comprimirla utilizando el estándar JPEG2000 se puede obtener una imagen con extensión *.j2k* con un tamaño de 64 KB. Para utilizar el compresor se digita en el símbolo del sistema el siguiente código:

```
java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaEncoder.j2k -rate 2
```

Estas líneas de comandos indican que se va a codificar el archivo de entrada *lena.pgm* localizado en la partición "C" del disco duro, obteniendo un archivo comprimido a una tasa de muestreo de 2bpm con extensión *.j2k* y guardado con el nombre de *lenaEncoder.j2k* en la partición "C".



```
Símbolo del sistema
C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaEncoder.j2k -rate 2
Target bitrate = 2.0 bpp <i.e. 65536 bytes>
Achieved bitrate = 1.9867859 bpp <i.e. 65103 bytes>
C:\>
```

La respectiva imagen obtenida es la que se muestra en la *figura 2*, la cual sigue teniendo una buena resolución.



Figura 2. LenaEncoder.j2k

Para detallar mejor y darnos una idea más clara acerca del nivel de compresión que proporciona el estándar JPEG2000, se mostrará la imagen de Lena comprimida a varios niveles de tasas de muestreo, tales como 3bpm, 2bpm, 1bpm, 0.1bpm, 0.01bpm y 0.001bpm para observar mejor el comportamiento. A continuación se muestra las líneas de comandos para cada tasa de muestreo.



```
Símbolo del sistema
C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaRate3.j2k -rate 3
Target bitrate = 3.00 bpp (i.e. 98304 bytes)
Achieved bitrate = 2.7911682 bpp (i.e. 91461 bytes)

C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaRate2.j2k -rate 2
Target bitrate = 2.00 bpp (i.e. 65536 bytes)
Achieved bitrate = 1.9867859 bpp (i.e. 65103 bytes)

C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaRate1.j2k -rate 1
Target bitrate = 1.00 bpp (i.e. 32768 bytes)
Achieved bitrate = 1.0116577 bpp (i.e. 33150 bytes)

C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaRate01.j2k -rate 0.1
Target bitrate = 0.10 bpp (i.e. 3276 bytes)
Achieved bitrate = 0.09991455 bpp (i.e. 3274 bytes)

C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaRate001.j2k -rate 0.01
Target bitrate = 0.010 bpp (i.e. 327 bytes)
Achieved bitrate = 0.009796143 bpp (i.e. 321 bytes)

C:\>java JJ2KEncoder -i C:\Lena.pgm -o C:\LenaRate0001.j2k -rate 0.001
[ERROR]: Overall target bitrate too low, given the current bit stream header
overhead

C:\>
```

Para una tasa de 0.001bpm el compilador genera un error debido que se ha bajado en exceso las muestras tomadas, se está tomando la milésima parte de un bpm.

El comportamiento de la tasa de muestreo con relación a la calidad del archivo está directamente relacionado. En la *figura 3* se muestra las respectivas graficas de cada una de las imágenes de Lena.

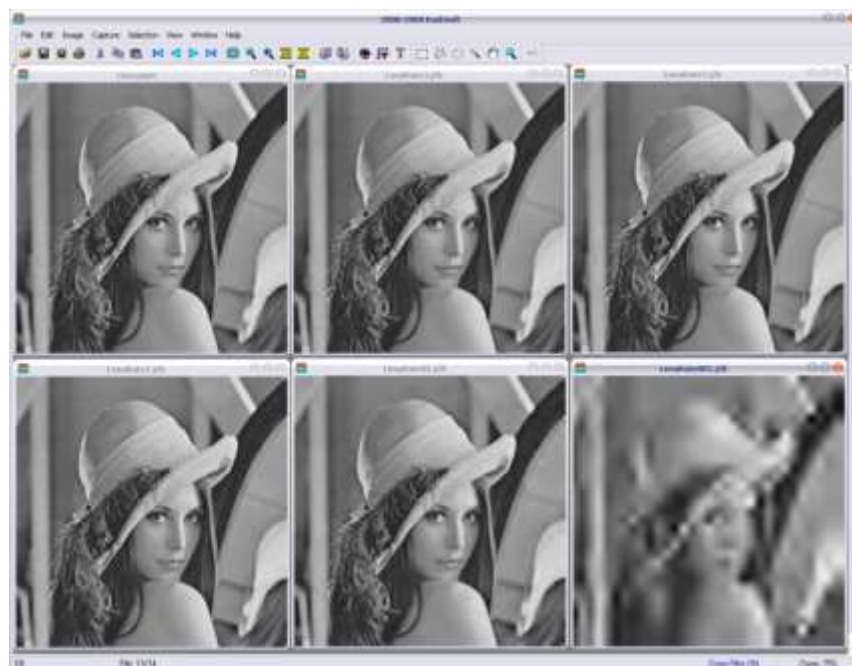


Figura 3. Imagen Lena Comprimida a diferentes tasas.



La anterior imagen de Lena esta en niveles de grises, ahora se tomará una imagen de Lena con extensión .ppm, con un tamaño de 769 KB. En la *figura 4* se muestra la imagen original.



Figura 4. Lena.ppm

La imagen comprimida a un nivel de tasa de 2 es la que se muestra en la *figura 5*.



Figura 5. LenaEncoder.j2k

La distorsión de los detalles, brillo y textura se van perdiendo a medida que se cambia la tasa de muestreo. A continuación se mostrará dicho comportamiento además de observar



mejor las características básicas que nos proporciona el estándar de compresión de imágenes JPEG2000 y el funcionamiento del JJ2000.

Utilizando las tasas de 2bpm, 1bpm, 0.1bpm, 0.05bpm, 0.01bpm. A continuación se mostrará las diferentes imágenes dependiendo de las tasas de compresión, en la *figura 6* se muestra la imagen de Lena comprimida a las tasas indicadas anteriormente.

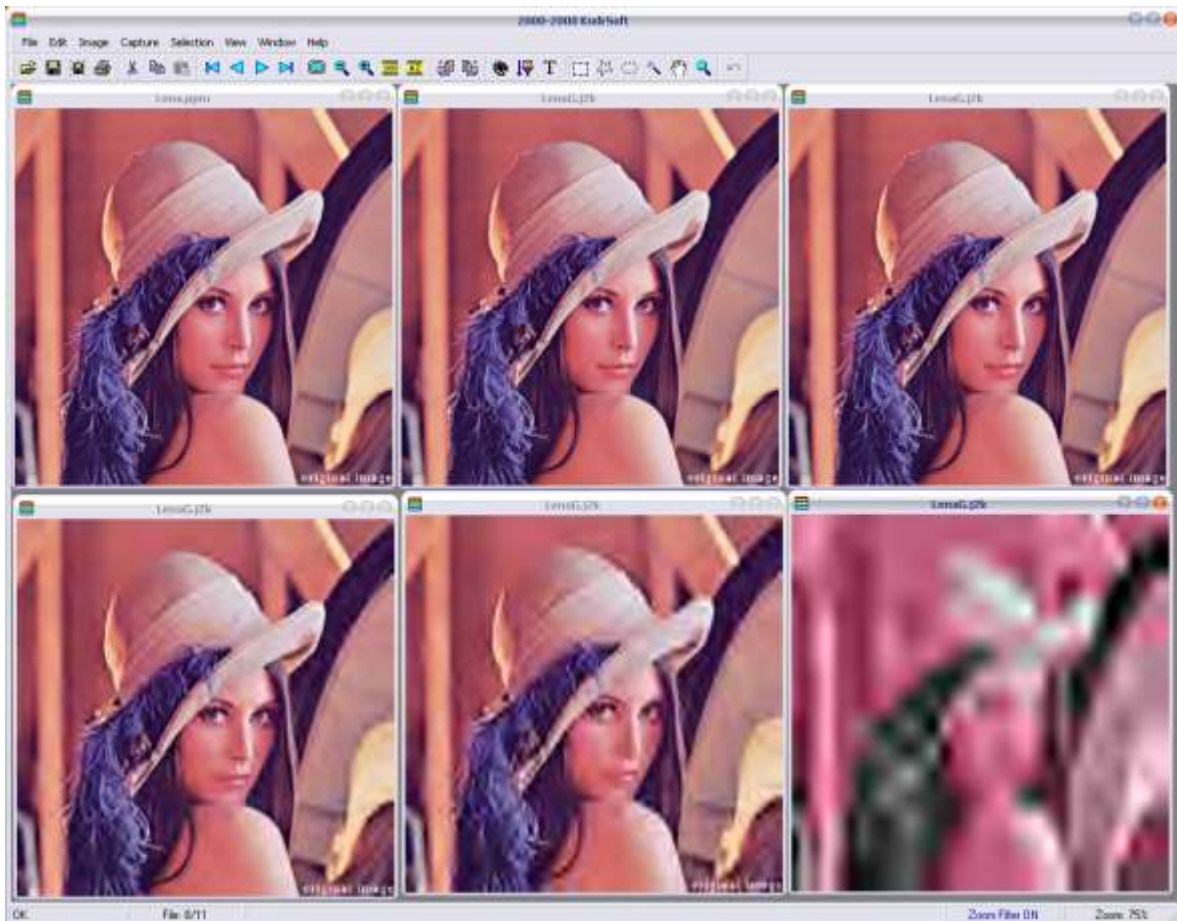


Figura 6. Imagen de Lena comprimida a diferentes tasas.

Con estas muestras se finaliza la fase de pruebas del JJ2000, concluyendo y verificando el correcto funcionamiento del código proporcionado.