Apéndice B

Funciones de librería

En este apéndice se va a explicar con detalle las funciones de las librerías JAI y JMF que se han utilizado. También se incluyen funciones pertenecientes a AWT.

B.1. Java Advanced Imaging, JAI

- **JAI.create**
  - Declaración 1: RenderedOp javax.media.jai.JAI.create(String arg0, Object arg1)
  - Parámetros:
    - String arg0: nombre de la operación.
    - Object arg1: objeto parámetro.
  - Descripción: Esta función crea una imagen como objeto RenderedOp.
    Si arg0="stream", obtiene la imagen del archivo donde se encuentra.
    Si arg0="scale", escala la imagen según los valores introducidos en el objeto arg1.
    Si arg0="AWTImage", sirve para obtener una PlanarImage a partir de una Image.
  - Declaración 2: RenderedOp javax.media.jai.JAI.create(String arg0, ParameterBlock arg1)

- Parámetros:
  - String arg0: nombre de la operación.
  - ParameterBlock arg1: objeto que contiene las fuentes y/o parámetros de la operación.
- Descripción: Crea una imagen a partir de una operación cuyos parámetros están contenidos en un objeto ParameterBlock. En la forma JAI.create(.ˆextrema", pb), permite obtener los valores extremos de las bandas de una imagen.

- Declaración 3: RenderedOp javax.media.jai.JAI.create(String arg0, ParameterBlock arg1, RenderingHints arg2)
- Parámetros:
  - String arg0: nombre de la operación.
  - ParameterBlock arg1: objeto que contiene las fuentes y/o parámetros de la operación.
  - RenderingHints arg2: condiciones de presentación.
- Descripción: Crea una imagen a partir de una operación cuyos parámetros están contenidos en un objeto ParameterBlock, y bajo una condiciones de representación dadas por arg2.
  Con arg0=“subtract” resta una imagen a otra.
  Con arg0=“mean”, se obtienen los valores medios de las bandas de una imagen.
  Si arg0=”BandCombine”, se combinan las bandas de una imagen multibanda según lo especificado en arg1. Con arg0=”histogram”, se calculan los histogramas de las bandas de una imagen.

- **PlanarImage.getAsBufferedImage**
  - Declaración: BufferedImage javax.media.jai.PlanarImage.getAsBufferedImage()
  - Descripción: Devuelve una copia de la imagen como BufferedImage.

- **RenderedOp.getProperty**
  - Declaración: Object javax.media.jai.RenderedOp.getProperty(String arg0)
  - Parámetros:
    - String arg0: propiedad que se pretende obtener.

- Descripción: Devuelve la propiedad arg0 de un objeto RenderedOp. Si arg0="extrema", se obtienen los valores máximos y mínimos de las bandas de la imagen representada por el objeto RenderedOp. Si arg0="mean", se obtienen los valores medios de las bandas.

- **PlanarImage.getProperty**
  - Declaración: Object javax.media.jai.PlanarImage.getProperty(String arg0)
  - Parámetros:
    - String arg0: propiedad que se pretende obtener.
  - Descripción: Devuelve la propiedad arg0 de un objeto PlanarImage. Si arg0="histogram", se obtienen los histogramas de las bandas de la imagen.

- **ROI.getAsImage**
  - Declaración: PlanarImage javax.media.jai.ROI.getAsImage()
  - Descripción: Devuelve una imagen binaria de tipo PlanarImage a partir de un objeto ROI.

- **DisplayJAI.set**
  - Declaración: void com.sun.media.jai.widget.DisplayJAI.set(RenderedImage arg0, int arg1, int arg2)
  - Parámetros:
    - RenderedImage arg0: Imagen que se va a presentar en la clase DisplayJAI.
    - int arg1: coordenada x de la imagen en el panel DisplayJAI
    - int arg2: coordenada y.
  - Descripción: Para colocar una imagen arg0 en un panel DisplayJAI, con coordenadas arg1 y arg2.

B.2. Java Media Framework, JMF

- **MediaLocator**
  - Declaración: javax.media.MediaLocator.MediaLocator(String arg0)
  - Parámetros:
    - String arg0: URL del archivo multimedia.
  - Descripción: Crea un MediaLocator a partir de una representación en String de una URL.
- **createProcessor**
  - Declaración: Processor javax.media.Manager.createProcessor(MediaLocator arg0) throws java.io.IOException, NoProcessorException
  - Parámetros:
    - MediaLocator arg0: objeto origen, describe el contenido multimedia.
  - Descripción: crea un objeto Processor a partir de un MediaLocator.
- **getData**
  - Declaración: Object javax.media.Buffer.getData()
  - Descripción: Obtiene el objeto interno con los datos de la porción del medio contenida en un Buffer.

B.3. Abstract Windowing Toolkit, AWT

- **PixelGrabber**
  - Declaración: java.awt.image.PixelGrabber.PixelGrabber(Image img, int x, int y, int w, int h, int[] pix, int off, int scansize)
  - Parámetros:
    - Image img: imagen origen, de la que se van a extraer píxeles.
    - int x, int y: coordenadas de los píxeles iniciales de una sección rectangular de img.
    - int w, int h: dimensiones de la sección rectangular.

- int[] pix: vector donde se almacenan los píxeles de la sección.
- int off: offset en el vector donde almacenar el primer píxel.
- int scansize:distancia de una fila de píxeles a la siguiente en el vector.

- Descripción: Crea un objeto PixelGrabber para obtener una matriz de una sección rectangular de píxeles (x, y, w, h) desde la imagen especificada al vector dado. Los píxeles se almacenan en el vector en el ColorModel RGB.

▪ textbfMemoryImageSource

- Declaración: java.awt.image.MemoryImageSource.MemoryImageSource(int w, int h, ColorModel cm, int[] pix, int off, int scan)
- Parámetros:
  - int w, int h: dimensiones de la imagen que se va a crear.
  - ColorModel cm: modelo de color especificado para la imagen.
  - int[] pix: vector donde se almacenan los píxeles de la sección.
  - int off: offset en el vector donde almacenar el primer píxel.
  - int scansize:distancia de una fila de píxeles a la siguiente en el vector.
- Descripción: Construye un objeto Image a partir de un vector de enteros. Realiza la operación inversa a PixelGrabber.