

Introducción a la librería CImg y puesta en funcionamiento

Captura y Procesamiento Digital de Señales e Imágenes

Departamento de Informática, FICH - UNL

26 de marzo de 2008

1. Introducción

Este documento presenta una reseña de las funciones fundamentales para cargar una imagen de archivo, obtener información de la misma, visualizarla y otras cuestiones de manejo básico, que contiene la librería CImg para procesamiento de imágenes en C++.

La librería está disponible en: <http://cimg.sourceforge.net>

Es necesario también tener instalado:

- un compilador de C++ (gcc),
- el paquete ImageMagick para la creación de las ventanas, disponible en <http://www.imagemagick.org>. Este paquete está preinstalado en varias distribuciones de Linux, para saber si ya está instalado ejecutar la instrucción `display` desde un terminal (debe mostrar la ventana del programa),
- la librería de manejo de las X en Linux: paquete `libX11-dev`.

2. Compilación

Durante la compilación con `g++` de un código fuente `prog.cpp` es necesario indicar las librerías enlazadas, de la siguiente manera:

```
g++ -o prog prog.cpp -lm -lpthread -lX11
```

Si se edita el archivo fuente con Anjuta, es posible dejar como parámetros de la compilación las librerías, indicando en el menú “Opciones/Opciones del compilador y enlazador”, solapa “Opciones”, campo “Flags del compilador (CFLAGS)”:

```
-L/usr/X11R6/lib -lm -lpthread -lX11.
```

3. Manejo básico de una imagen

- Crear una imagen vacía:

```
CImg<tipo> img(p1,p2,p3,p4,p5);
```

El tipo de dato utilizado para guardar los valores de la imagen es variado: `unsigned char`, `int`, `float`, ... El tipo por defecto es `float`, accesible mediante la instrucción `CImg<> img...`

Los parámetros indicados con `px` son:

- `p1`: ancho, número de columnas.
- `p2`: alto, número de filas.
- `p3`: profundidad, número de cuadros.
- `p4`: dimensión del pixel, número de canales.
- `p5`: valor de inicialización de los pixeles, ej: 0 (negro).

Ejemplos:

- `CImg<unsigned char>img(640,480)`: imagen en tonos de grises.
- `CImg<unsigned char>img(640,480,1,1,0)`: idem anterior, pero inicialmente negra.
- `CImg<>img(3,3,1,1,1/9)`: imagen de 3x3 en tonos de grises inicializada con valores 1/9 (utilizada como máscara para filtrado de promediado).
- `CImg<int>img(200,100,1,3)`: imagen color.

- Crear una imagen cargándola de un archivo:

```
CImg<tipo> img("archivo.ext");
```

La instrucción anterior equivale a:

```
CImg<tipo> img;  
img=load("archivo.ext");
```

- Campos de la estructura `CImg`: accesibles a través de las siguientes funciones, mediante la instrucción `img.funcion`:

- `dimx()`: ancho, número de columnas.
- `dimy()`: alto, número de filas.
- `dimz()`: profundidad, número de cuadros.

- `dimv()`: dimensión del pixel, número de canales.
- `ptr()`: datos, puntero a la matriz imagen.

La matriz se accede desde el elemento (0,0) en la esquina superior izquierda, hasta el elemento (ancho-1,alto-1) en la esquina inferior derecha.

■ Visualización de una imagen:

La matriz de imagen se visualiza en una estructura de datos especial que crea la ventana y además maneja eventos de teclado y mouse: `CImgDisplay ventana(img,titulo",n);`

El parámetro `img` es la matriz con la imagen propiamente dicha, mientras que el parámetro `n` es la normalización de la visualización (0: sin normalización al intervalo [0,255], 1: normalizada ó 2: cambia el comportamiento por defecto a normalizada de ahí en adelante)¹. La cadena de caracteres "titulo" fija el título que aparecerá en la barra superior de la ventana.

El ejemplo siguiente muestra la creación de la ventana y un bucle de espera necesario que espera la pulsación de la tecla ESC o Q para terminar la ejecución:

```
CImgDisplay ventana(img,titulo",0);
while (!ventana.is_closed && ventana.key!=cimg::keyQ) { }
```

Cambia/vuelve a pantalla completa:

```
ventana.toggle_fullscreen();
```

Un comportamiento especial es considerado cuando se debe sobreimprimir alguna información o gráfica sobre la imagen, por ej., para mostrar un cuadro con las coordenadas del puntero del mouse.

En estos casos, en el bucle de espera se re-dibuja la ventana mediante una re-definición de la estructura donde los paréntesis entre el nombre de la imagen hacen referencia al contenido previo (de esta manera no se pierde la imagen cargada). Las órdenes de graficación extras y una llamada a la visualización mediante una tubería se muestra en la línea siguiente²:

```
CImg<tipo> (img).draw_line(...).display(ventana);
```

■ Grabación de una imagen:

Para guardar en disco una imagen, simplemente se llama a la función: `img.save("nombre.ext");`

¹Más información en página 13 del manual de referencia.

²La llamada a `draw_line()` es solamente un ejemplo de cualquier función utilizable.

4. Funciones adicionales

- Agregar un comentario de ayuda al invocar el programa mediante la línea de comandos con la opción -h:
`cimg_usage("Este programa realiza...");`

5. Comentarios finales

Este documento es de libre distribución y reproducción total o parcial por cualquier medio.

Comentarios y sugerencias: `cmartinez@fich.unl.edu.ar`