Xv, ImageMagick y Gimp, Tres Programas para la Visualización y Edición de Imágenes

Luis Gerardo de la Fraga

Sección de Computación
Departamento de Ingeniería Eléctrica. CINVESTAV-IPN.
Av. Instituto Politécnico Nacional 2508. 07300 México, D.F.
E-mail: fraga@cs.cinvestav.mx

Resumen

En este trabajo se describirá brevemente como manipular imágenes con los programas *xv, ImageMagick* y *gimp* del sistema operativo GNU/Linux. El objeto del trabajo es enseñar las herramientas básicas para la edición y manipulación de imágenes que podrían ser incluidas en nuestras presentaciones y particularmente en documentos LaTeX. Por último se realiza una animación usando las herramientas básicas de GNU/Linux y el programa *gimp*.

1 Introducción

La edición de imágenes y su inclusión en documentos de texto es una tarea muy común hoy en día, en un mundo en el que las imágenes predominan como el medio principal para la transmisión de ideas. Un ejemplo inmediato lo vemos en el diseño de las páginas que se ponen disponibles en la WEB. Es por ello que resulta importante conocer con que programas contamos en GNU/Linux para la edición de imágenes y como podemos incluirlas en los documentos que realizamos.

En este artículos nos enfocaremos a tres tareas básicas:

1. Cómo crear o capturar imágenes nuevas.

2. Cómo editar imágenes para agregarles etiquetas y texto, y

3. Cómo realizar animaciones.

Los tres programas que veremos son *xv, ImageMagick* y *gimp*. De entrada para usar estos programas es muy importante contar con un ratón de tres teclas, ya que todos estos programas lo usan. Si sólo se cuenta con un ratón de dos teclas hay que habilitar la emulación de tres botones cuando se configura el sistema X (el botón central se emula presionando los dos botones al mismo tiempo). Como "sistema X" se conoce la suite de programas que nos permite trabajar en un ambiente gráfico en GNU/Linux.

2 El programa xv

Xv permite visualizar imágenes en varios formatos, tales como GIF, JPEG, TIFF, PBM, PGM, PPM, X11 bitmap, etc. (vea el manual que queda instalado en el archivo /usr/doc/xv-3.10a/xvdocs.ps, o las páginas man).

Este programa se distribuye en dos archivos RPMS en la versión 7.x de RedHat Linux: xv-3.10a-23.i386.rpm y xv-docs-3.10a-23.i386.rpm, y forman parte de la distribución de PowerTools. Antes de la llegada de las distribuciones por medio de rpm (Red Hat Packet Manager), había que bajar el código fuente de Internet, adecuarlo a nuestra máquina Unix, compilarlo e instalarlo. Ahora esto es trivial con el comando `rpm -i %.rpm`, para instalar un paquete `%.rpm`.

Xv es un famoso programa escrito por John Bradley y el sitio primario para obtener el código fuente es ftp.cis.upenn.edu en el directorio pub/xv. Un artículo que habla sobre xv está en la referencia [1]. xv también puede usarse para capturar imágenes en

GNU/Linux por medio de un scanner [2]. Xv tiene el inconveniente de que no es libre (freeware), es *shareware*, esto es, libre para uso personal. El costo de registro es de 25 USD y están disponibles licencias institucionales.

Para ejecutarlo simplemente se puede teclear `xv` en la línea de comandos, y aparecerá la "imagen logo" de xv. Para visualizar una imagen se ejecuta `xv nombre_imagen`. La ventana principal de comandos de xv aparece al hacer clic con el botón izquierdo del ratón sobre la imagen desplegada por xv y puede verse en la Fig. 1

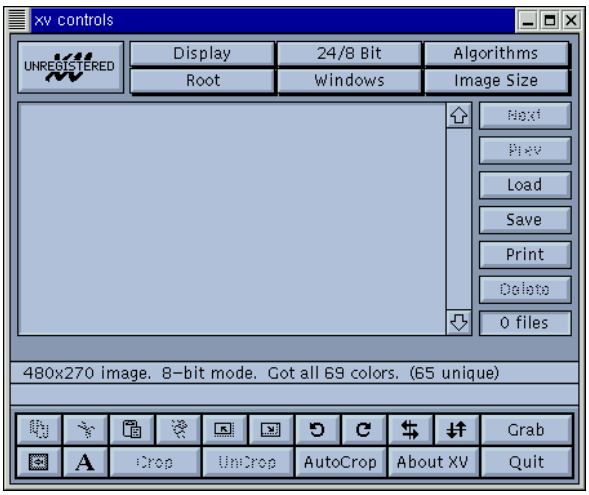

Figura 1: La ventana de control de xv que se despliega al hacer clic con el botón izquierdo sobre la imagen desplegada.

Una característica de xv muy útil es la capacidad de capturar una ventana o trozos de la pantalla en una imagen, para ello debemos utilizar el botón Grab que puede verse en la parte inferior derecha de la Fig. 1. Es posible usar xv totalmente a través del teclado haciendo uso de las teclas rápidas (cada tecla tiene asignada una función, la ventana de control se abre precionando la tecla '?'). Al hacer clic sobre el botón de Grab sale otra ventana en la que se puede introducir un tiempo de retardo después del cual, si se oprime la tecla izquierda del ratón se graba una ventana, si se mantiene oprimida la tecla central se graba un área rectangular que puede abarcar toda la pantalla, y si se oprime la tecla derecha se cancela la acción de grabar.

Otra utilidad importante es la ventana de edición de colores, Windows → Color Editor (u oprimiendo la letra 'e' con el cursor sobre la imagen) abre esta ventana. Aquí se pueden aplicar una normalización y/o igualación del histograma, que son dos técnicas de procesamiento de imagen muy útiles cuando se tiene una imagen con un contraste muy pobre; así como también se puede modificar el color asociado a los pixels de la imagen.

Por último, es muy fácil cortar un trozo de imagen seleccionando una región rectangular, manteniendo oprimido el botón derecho del ratón sobre la imagen, y oprimiendo después el botón Crop (o la tecla 'c'). Las medidas exactas de la región de corte pueden verse si se despliega la ventana de información de la imagen Windows → Image Info (o la tecla 'i').

3 El paquete ImageMagick

Para instalar ImageMagick son necesarios los siguientes archivos RPM ImageMagick-5.2.7-2-.i386.rpm, ImageMagick-devel-5.2.7-2.i386.rpm, ImageMagick-c++-5.2.7-2.i386.rpm, e ImageMagick-c++-devel-5.2.7-2.i386.rpm para la versión de RedHat 7.1. Estos archivos forman parte de la distribución estándar de RedHat.

ImageMagick es un paquete de programas en X11 para la visualización y manipulación de imágenes. Fue desarrollado por DuPont y el sitio primario para obtener el código fuente es ftp://ftp.wizards.dupont.com/pub/Image-Magick/

ImageMagick está compuesto por nueve programas: *display*, *import*, *animate*, *montage*, *convert*, *mogrify*, *identify*, *combine* y *xtp*. Los que usaremos son *display*, para desplegar y editar interactivamente una imagen; *convert*, para conversión entre distintos formatos; y *montage* para combinar varias imágenes. El programa *import* nos puede servir para capturar una ventana o toda la pantalla. La documentación de ImageMagick está en formato HTML en el directorio /usr/share/doc/ImageMagick-5.2.2, donde puede verse toda la documentación de cada programa o en las páginas man.

Vamos ahora a realizar algunas operaciones con una imagen. Supongamos que queremos agregarle texto. Para ello, a una imagen ya creada (puede hacerse grabando una parte de la pantalla con xv), se realiza:

1. Abrir la imagen dando en la línea de comandos `display nombre_imagen`.

2. Haciendo clic con el botón izquierdo del ratón, se despliega la ventana del menú de ImageMagick. Con Image Edit → Annotate se despliega el menú para agregar anotaciones. Se selecciona el tipo de font, su color, el color de fondo (Box Color) y si es un texto rotado; finalmente se hace clic con el botón izquierdo del ratón sobre la imagen en la posición que queremos introducir el texto.

3. Se repite el paso 2 para cada etiqueta de texto que queramos introducir en la imagen.

Si las etiquetas de texto las queremos poner sobre un borde fuera de la imagen, hay que hacer Image Edit → Add Border. . . , luego seleccionamos el color del borde, y el tamaño de la geometría del borde (por defecto el tamaño es de 6×6 pixels). Podemos quitar un borde que no queramos con la operación Transform → Chop.

Para agregar alguna figura geométrica, una línea, circulo, circunferencia, etc., se realiza Image Edit → Draw. . . y se abre el menú de dibujo. Su uso es intuitivo, por lo que no se describirá aquí.

Las "operaciones rápidas" (short cuts) se despliegan si se hace clic sobre la imagen con el botón derecho del ratón. Y así podemos desplegar la información del imagen, deshacer una operación (undo) o salvarlo con otro formato (ó hacer File → Save sobre el menú principal).

Un uso avanzado de ImageMagick, muy bien descrito en la documentación, es el uso de sus funciones de manipulación de imagen a través de un lenguaje de programación como PERL [3], C, ó C++.

4 El programa gimp

Gimp es el Programa Manipulador de Imágenes de GNU (GNU Image Manipulation Program). Gimp ha sido diseñado pensando en contar con un programa libre profesional para manipular imágenes. Para saber más sobre el proyecto GNU visite www.gnu.org

En RedHat 7.1, la distribución completa de gimp consta de cuatro archivos RPM: gimp-1.2.-1-5.i386.rpm, gimp-data-extras-1.2.0-1.noarch.rpm gimp-devel-1.2.1-5.i386.rpm y gimp-perl-1.2.1-5.-i386.rpm; aunque los dos últimos son necesarios sólo para programadores.

El sitio primario de información para gimp es www.gimp.org y existe un excelente manual en línea en manual.gimp.org . El uso de *gimp* es complejo y se convida referirse al manual para apoyarse en su uso.

Vamos a ver un ejemplo de como crear un GIF animado usando como ejemplo el juego LIFE de vida artificial. El juego LIFE fue creado por John Conway en 1970 [4] y consiste de una serie de células, cada una de las cuales tiene ocho células vecinas. Una célula se considera ocupada (por un organismo) o desocupada. Las reglas para derivar una generación son las siguientes: Si una célula ocupada tiene 0, 1, 4, 5, 6, 7 ú 8 células vecinas ocupadas, el organismo *muere* (0, 1, muere por aislamiento; 4-8 por sobrepoblación); si una célula ocupada tiene dos o tres vecinas ocupadas, el organismo *sobrevive* a la siguiente generación; si una célula desocupada tiene tres vecinas ocupadas, la célula *nace*, esto es, se convierte en ocupada. Una búsqueda en Internet en cualquier buscador, p.e. www.google.com con la clave "game life scientific american" nos dará cientos de referencias sobre LIFE.

En el apéndice se presenta el código en PERL para el juego LIFE. En el código, cada célula corresponde a un pixel sobre una imagen y las células vecinas a los ocho pixels vecinos a él[5, p. 40]. Los datos necesarios para su ejecución son el tamaño de las imágenes, el número de generaciones y la población (imagen) inicial, que se pueden editar al inicio del código. Se incluye una función para guardar una imagen (que representa a una generación, ó un marco [*frame*] de nuestra animación) en formato PBM legible por cualquiera de los programas descritos aquí. El listado puede obtenerse en la dirección http://delta.cs.cinvestav.mx/˜fraga/Programas/life.pl

Para ejecutar el programa hay que volver el archivo ejecutable y ejecutarlo propiamente:

```
chmod 700 life.pl
./life
```

Obtendremos de salida diez imágenes cuyos nombres tienen el formato `t_????.pbm`, donde `????` va de 0001 a 0010 (por tener la variable `$generations` definida igual a 10).

Las imágenes de salida son muy pequeñas y para escalarlas podemos usar el programa en PERL de la Fig. 2.

```
#!/usr/bin/perl

@files = ‘ls -1 *.pbm‘;
# Tenemos los nombres en el arreglo @files
$n = @files;   # El número de imágenes

for ( $i=0; $i<$n; $i++ ){
    $name = sprintf ( "a_%04d.tif", $i );
    $input  =  $files[$i];
    chop ( $input );
    print "E: -scale 64  $input $name\n";
    ‘convert -scale 64 $input gray:tmp‘;
    ‘convert -size 64x64 gray:tmp $name‘;
}
```

Figura 2: El programa en PERL usado para escalar las imágenes para nuestra animación

Para el programa de la Fig. 2, el comando `chop ( $input )` quita el último caracter a cada nombre de la imagen de entrada. Se quita este caracter porque representa al retorno de carro ‘`\n`’. Se observa también sobre la Fig. 2 que se utiliza el programa *convert* de ImageMagick para escalar las imágenes de entrada al nuevo tamaño de 64×64 pixels.

Gimp puede realizar un GIF animado solamente con imágenes en tonos de gris o en color verdadero RGB (en estas imágenes cada pixel está representado por su tripleta de componentes de color rojo, verde y azul), es por ello que se utiliza dos veces el programa *convert* en la Fig. 2, una para crear una imagen temporal ya escalada en tonos de gris y luego cambiamos el formato de la imagen temporal a una imagen TIF cuyo nombre tendrá el formato `a_????.tif` donde ‘????’ tendrá la misma numeración que las imágenes `t_????.pbm` de salida del programa life.pl. Este formato define un marco para gimp.

Para realizar ahora la animación con gimp es muy sencillo. Ejecutamos gimp y abrimos la primera imagen `a_0001.tif`. Y sobre esta imagen agregamos todo los demás marcos: clicamos con el botón derecho del ratón sobre la imagen abierta y usamos la opción Video → Frames to Image; aquí se abre una nueva ventana donde podemos editar el tiempo de exposición de los marcos (por defecto es 41 milisegundos), para nuestro ejemplo un valor razonable es del 800 mseg. La animación creada es mostrada por gimp en una nueva ventana; clicando son el botón derecho del ratón sobre esta ventana podemos ver la animación con la opción Filters → Animation → Animation Playback. Finalmente, para salvar la animación creada usamos la opción File → Save, se abre una nueva ventana donde debemos poner el nombre del archivo de salida y escogemos el formato de salida como GIF. La animación final puede visualizarse también con cualquier visor de WEB (netscape, OutLook, etc.).

En el caso de que se hayan creado cada marco como imágenes en color, deben salvarse como imágenes en color verdadero o RGB. Una vez que se han agregado todos los marcos, hay que cambiarlos a imágenes indexadas usando Image → Mode → Indexed. De esta manera es posible salvar la animación en color como un gif animado.

Es posible también visualizar todos los marcos con *xv* o con *animate* (que es parte de ImageMagick). Con xv se usaría el comando:

```
xv -wait 1 -wloop a_00??.tif
```

donde la opción `-wait 1` es para mostrar cada imagen por un intervalo de 1 segundo, y la opción `-wloop` es para que entre en un ciclo donde muestre continuamente todas la imágenes al terminar con la última.

Para visualizar todos las imágenes con el programa *animate* de ImageMagick se haría simplemente:

```
animate a_00??.tif
```

y clicando con el botón izquierdo sobre la imagen abierta se puede controlar la velocidad de exposición de cada imagen con la opción Speed → Faster y Speed → Slower.

Nueve generaciones del juego LIFE, de salida del programa life.pl en el apéndice, se muestran en la Fig. 3. Esta imagen fue generada con el programa *montage* de ImageMagick con las opciones:

```
montage -tile 3x3 -geometry 16x16 *.pbm a.tif
```

Se invita a ver las páginas man de *montage* para la explicación de cada una de estas opciones.

5 Conclusiones

Se han presentado tres programas para la edición y creación de imágenes en el sistema operativo

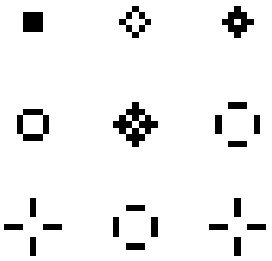

Figura 3: Nueve generaciones del juego LIFE tal como se obtienen al ejecutar el programa life.pl mostrado en el Apéndice. Se forma un patrón "estable", que puede comprobarse al aumentar el número de generaciones sobre el programa life.pl

GNU/Linux: xv, ImageMagick y gimp. Xv e ImageMagick vienen con una muy buena documentación desde la instalación de ellos. Gimp tiene un excelente manual en línea en la dirección http://manual.-gimp.org.

Se ha presentado la creación de animaciones con gimp, usando como ejemplo el juego LIFE, programado en el lenguaje de muy alto nivel PERL. Gimp es un paquete para uso profesional de edición de imágenes en el sistema GNU/Linux. Debido a que en cada nueva versión de gimp, las opciones y posibilidades del programa cambian continuamente, se recomienda leer la documentación para profundizar en sus opciones y sus capacidades.

Las opciones que ofrecen ImageMagick y gimp para edición de imágenes pueden usarse también desde programas en PERL, C ó C++. Esto representa programación avanzada y se recomienda dirigirse a la documentación si se está interesado en este modo de usarlos.

En general se recomienda dirigirse a la documentación de estos programas para profundizar en su uso, ya que aquí solo ha realizado una muy breve introducción al uso de ellos, con el fin de motivar a usarlos en este mundo donde la transmisión de ideas se realiza a través de imágenes, y con la ventaja de que no debemos de pagar por ellos.

Referencias

[1] M. L. Richardson. xv: the X viewer. *Linux Journal*, 73, 2000. it can be reached at www.linux-journal.com.

[2] M. Richardson. XVScan. *Linux Journal*, 74, 2000. it can be reached at www.linux-journal.com.

[3] J. K. Ousterhout. Scripting: Higher-level programming for the 21st century. *IEEE Computer*, pages 23–30, 1998.

[4] M. Garder. Mathematical games. the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223(4):120–123, 1970. available at http://www.ibiblio.org/life-patterns/october1970.html.

[5] Gonzalez R.C and R.E Woods. *Digital Image Processing*. Adisson-Wesley, 1992.

Apéndice

```
#!/usr/bin/perl

# Programa en PERL para generar en imagen PBM
# una generación del juego LIFE

# Las dimensiones de la imagen de salida
$rows = 16;
$cols = 16;
$generations=10; # El número de generaciones,
                 # una imagen por generación

# La imagen inicial
for ( $i=0; $i<$rows; $i++ ) {
   for ( $j=0; $j<$cols; $j++ ) {
      $img[ $cols * $i + $j ] = 0;
   }
}
$n = $rows/2;
for ( $i=$n-2; $i<=$n; $i++ ) {
   for ( $j=$n-2; $j<=$n; $j++ ) {
      $img[ $cols * $i + $j ] = 1;
   }
}

# La realización del juego LIFE:
for ( $g=1; $g<=$generations; $g++ ) {
   $name = sprintf( "t_%04d.pbm", $g );
   print "Se guarda imagen $name\n";
   save_pbm_img ( \@img, $rows, $cols, $name );
```

```
   for ( $i=0; $i<$rows; $i++ ) {
      # Consideramos a la imagen un toro
      $y1 = ($i-1) % $rows;
      $y2 = ($i+1) % $rows;

      for ( $j=0; $j<$cols; $j++ ) {
         $x1 = ($j-1) % $cols;
         $x2 = ($j+1) % $cols;

         $suma = 0;
         $suma += $img[ $cols * $y1 + $x1 ];
         $suma += $img[ $cols * $y1 + $j ];
         $suma += $img[ $cols * $y1 + $x2 ];
         $suma += $img[ $cols * $i  + $x1 ];
         $suma += $img[ $cols * $i  + $x2 ];
         $suma += $img[ $cols * $y2 + $x1 ];
         $suma += $img[ $cols * $y2 + $j ];
         $suma += $img[ $cols * $y2 + $x2 ];

         if ( $img[ $cols * $i + $j ] > 0 ) {
            if( $suma <= 1 || $suma >= 4 ) {
               # Muere el bicho
               $imgout[ $cols * $i + $j ] = 0;
            }
            else { # Vive el bicho
               $imgout[ $cols * $i + $j ] = 1;
            }
         }
         elsif( $suma == 3 ) {    # Nace uno!
            $imgout[ $cols * $i + $j ] = 1;
         }
      }
   }
   # Copiamos la imagen @imgout a @img
   for ( $i=0; $i<$rows*$cols; $i++ ) {
      $img[$i] = $imgout[$i];
   }
}

# Subrutina para salvar una imagen en formato PBM
# crudo (Portable Bitmap Map, ver la página man de
# "pgm" sobre el paquete NETPBM)
# Entrada: ( @img, $rows, $cols, $filename )
sub save_pbm_img (\@$$$) {
   local (*myimg) = shift(@_);
   my $mycols = shift(@_);
   my $myrows = shift(@_);
   my $filename = shift(@_);

   open ( MYFILE, ">$filename" ) or return 1;
   select MYFILE;   $| = 1;

   print MYFILE "P4\n";
   print MYFILE "# file: $filename\n";
   print MYFILE "$mycols $myrows\n";

   my $data; my $n = int( ($mycols + 7)/8 )*8;
   for ( my $i=0; $i<$myrows; $i++ ){
      $data = "";
      for ( my $j=0; $j<$mycols; $j++ ){
         if ( $myimg[ $mycols * $i + $j] ) {
            $data = $data.'1';
         }
         else { $data = $data.'0'; }
      }
      my $bytes = pack( "B$n", $data );
      syswrite ( MYFILE, $bytes );
   }
   close MYFILE;
   select STDOUT;
   return 0;
}
```