

Panorámicas perfectas con Hugin, Enblend y Autopano-SIFT

Vistas de 360 grados



Podemos crear una imagen agradable alineando múltiples paisajes para conformar un panorama. Pero la alineación puede ser complicada, especialmente si las fotografías individuales no coinciden. Herramientas como Hugin, Enblend y Autopano-SIFT pueden ayudarnos a crear panorámicas perfectas sin marcas.

POR OLIVER FROMMEL

Si tenemos más fotos de las que podemos ajustar en una imagen puede que queramos intentar el viejo truco de combinar múltiples fotografías digitales en una agradable vista panorámica. Será muy difícil combinar estas fotografías en una imagen sin cortes usando Gimp. Este artículo trata algunas de las herramientas alternativas que podemos usar para ajustar y alinear las imágenes automáticamente. Estas herramientas crearan panorámicas sin cortes partiendo de fotografías separadas.

Es fácil tomar una serie de fotografías para formar una panorámica, pero aquí es donde comienza el problema. Es difícil conseguir cortes suaves e invisibles. Normalmente, las fotografías están tan deformadas que es imposible corregirlas manualmente. Las lentes son parcialmente culpables. Éstas distorsionan normalmente las imágenes, convirtiendo líneas rectas en curvas que hacen la composición de fotografías difícil.

La otra razón es la distorsión causada por la propia realización de una serie de fotografías. Si permanecemos quietos y

giramos la cámara de izquierda a derecha, los objetos de la derecha de la imagen aparecerán muy grandes, mientras que los de la izquierda serán más pequeños.

Hay software gratuito que nos puede ayudar a corregir este tipo de distorsión, alinear imágenes individuales y finalmente montarlas para crear una panorámica.

Ecuilibración y Alineado de Fotografías

El programa de gráficos Hugin [2] muestra las imágenes permitiendo al usuario seleccionar puntos neurálgicos para ecualizar y alinear las imágenes.

El programa es fácil de instalar pero requiere algunas librerías que algunas distribuciones no incluyen. En el recuadro “Instalación” hay más detalles al respecto de la instalación.

Tras escribir *hugin* para lanzar el programa aparece la ventana mostrada en la figura 2. Pulsamos **Add individual**

Cuadro 1: PanoTools

Durante mucho tiempo, el paquete PanoTools fue el único software gratuito para crear imágenes panorámicas. Las condiciones de la licencia de PanoTools siempre han sido complejas. La librería y PTOptimizer se distribuye bajo la GPL, mientras que se aplican varias licencias a algunos de los programas.

Debido a que el autor del programa, Helmut Dersch, ya no tiene el programa en su página Web, otras personas ofrecen paquetes antiguos. De hecho hay incluso algunas versiones comerciales, si bien puede que no sean del todo legales. Se ha fundado un proyecto Sourceforge para mantener las partes gratuitas de PanoTools [1].

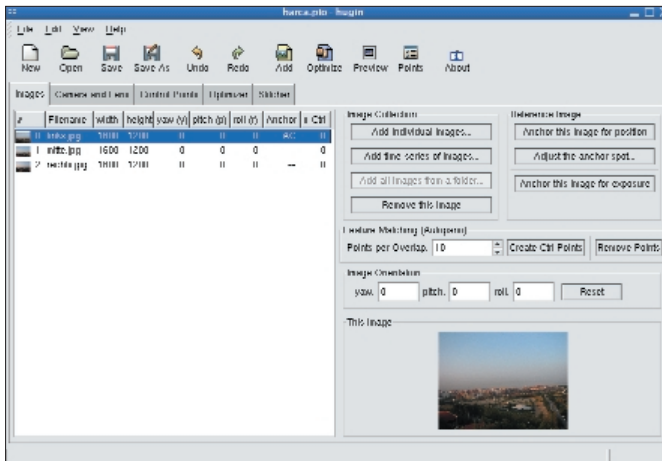


Figura 1: Carga de las imágenes individuales en la ventana principal de Hugin.

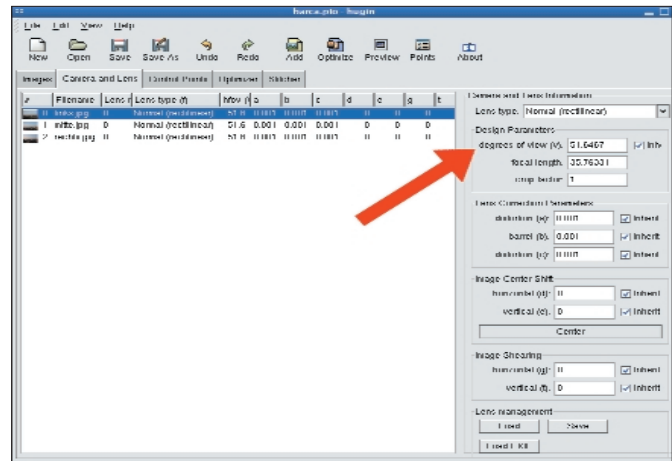


Figura 2: La pestaña Camera and Lens.

imágenes... para seleccionar imágenes individuales. En el diálogo que aparece podemos mantener pulsado [Ctrl] y pulsar para seleccionar múltiples imágenes.

Hugin muestra una lista de imágenes en el panel de la izquierda añadiendo algunos detalles como la anchura y altura o los valores de rotación (**Yaw**, **Pitch**, **Roll**), si bien estos cuentan con el valor cero por defecto. Hugin usa la primera fotografía como punto de referencia para alinear las otras imágenes. El botón "Anchor this image ..." permite seleccionar otra imagen como punto de referencia.

La segunda pestaña, *Camera and Lens*, se usa para corregir las distorsión

óptica causada por la cámara y las lentes. Si nuestras imágenes incluyen información EXIF, Hugin puede automáticamente extraer los datos requeridos. Si no conocemos el ángulo de la vista de nuestras lentes debemos consultar la Tabla 1 (extraída de la documentación de PanoTools).

El conmutador *Inherit* permite que el resto de las imágenes adquieran los ajustes de la imagen actual. Si deshabilitamos estos ajustes el programa puede gestionar imágenes con diferentes valores de distorsión. Tras descubrir los mejores valores para nuestra cámara (mediante el método de prueba y error)

podemos almacenar estos valores en *Lens Management* y cargarlos para nuestros próximos proyectos panorámicos.

Puntos de control manual

Hugin necesita unos detalles definitivos de las imágenes para ser capaz de alinearlas correctamente. La pestaña *Control Points* se encarga de esto. La pestaña nos muestra las dos imágenes permitiéndonos pulsar sobre los puntos de control para crear asignaciones.

Primero debemos ajustar Hugin para que muestre dos imágenes adyacentes ajustando el número en la pestaña sobre la imagen de la derecha un número más

Cuadro 2: Instalación

Hugin necesita la librería WxWidgets (antiguamente WxWindows) y la extensión WxWidgets-XRC. La mayoría de las distribuciones incluyen las otras librerías requeridas, como Libtiff, LibPNG y LibJPG. Al margen del paquete matemático FFTW, Hugin también necesita la librería Boost Library si pretendemos construir Hugin nosotros mismos.

Desafortunadamente, la versión y nombre dependen de la distribución. Por ejemplo, la librería Boost de desarrollo para Fedora simplemente se llama *libboost-devel*; los usuarios de Debian necesitan tres paquetes llamados *libboost-graph-dev*, *libboost-dev* y *libboost-python-dev*. Los archivos requeridos para Fedora están localizados en el subdirectorio del DVD. El siguiente comando instala las librerías:

```
rpm -iv panorama-tools-2.7a-1.i386.rpm wxGTK-* fftw2-*
```

Las herramientas Panorama incluyen `libpano12`, lo que no impide que el paquete gestor se queje de que el paquete falta porque las dependencias del paquete son erróneas. El camino que resuelve el problema implica especificar una opción del programa RPM para deshabilitar la comprobación de dependencia:

```
rpm -iv --nodeps hugin-0.4-cvs20041021.bp.fc2.i386.rpm
```

Si bien Suse tiene una versión de WxWindows por defecto, Hugin no puede usar la versión por defecto. Por tanto deberemos eliminar el paquete existente y usar los paquetes de Fedora en su lugar. No os preocupéis, funcionó bien en Suse 9.1 en nuestro laboratorio:

```
rpm -iv panorama-tools-2.7a-1.i386.rpm wxGTK-xrc-2.4.2-2-4.1.fc2.dag.i386.rpm fftw-2.1.3-3-1102.i586.rpm hugin-20040921-1.bp.fc1.i386.rpm wxGTK-2.4.2
```

2-4.1.fc2.dag.i386.rpm
Debian no dispone de los paquetes adicionales wxGTK que incluyen el nombre "xrc" en el paquete; en su lugar debemos buscar *libwxgtk2.4-contrib-dev*. Podemos usar los siguientes comandos para instalar las librerías requeridas, LibPano y Hugin:

```
apt-get install libwxgtk2.4-dev libwxgtk2.4-contrib-dev fftw2 dpkg -i libpano12_20040917-1_i386.deb dpkg -i hugin_0.4pre20040917-1_i386.deb
```

Si esto funciona o no depende del software y las librerías que hayamos instalado con anterioridad. Si aun nos falta algún paquete puede que queramos intentar buscarlo en Google por su nombre y formato. Los cuadros de Autopano y Enblend nos ofrecen más detalles sobre los requisitos

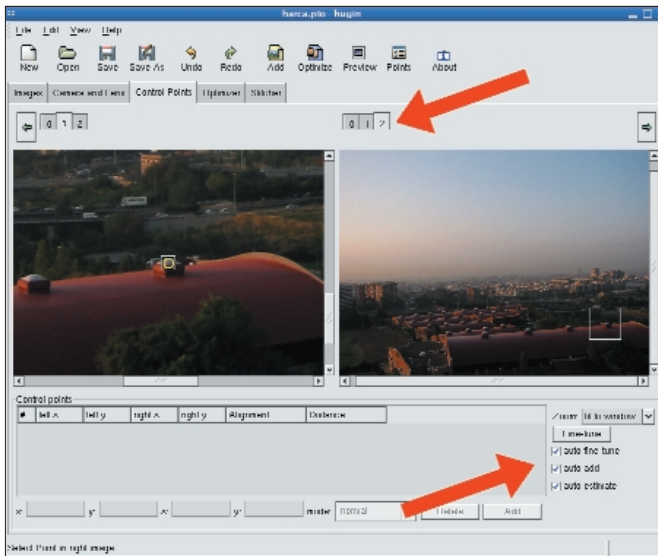


Figura 3: Hugin muestra dos imágenes adyacentes, permitiendo al usuario seleccionar los detalles de la imagen que se usarán como puntos de control.

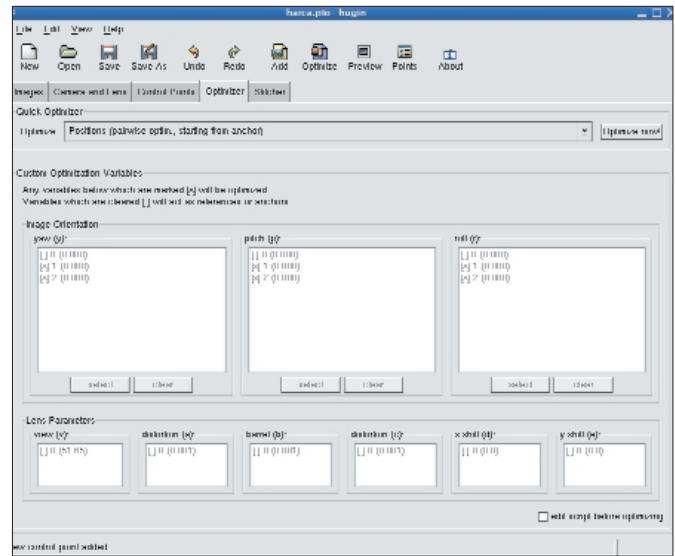


Figura 4: La pestaña *Optimizer* soporta el control granular de los valores de optimización ajustados automáticamente por Hugin.

alto que el de la pestaña sobre la imagen de la derecha (figura 3). En otras palabras, si cuando empezamos las dos imágenes tienen un cero cuando empezamos debemos ajustar la pestaña de la derecha a uno. Entonces podemos usar las teclas de las flechas para mover las imágenes y Hugin nos mostrará imágenes consecutivas a la izquierda y a la derecha.

También debemos echar un vistazo a *auto fine-tune* (ajuste fino automático) y *auto-estimate* (estimación automática), que le indica a Hugin que nos debe ayudar a asignar los puntos de control. La función *auto add* (añadir automáticamente) indica al programa que añada cada par de trazas en los puntos especificados, lo que también podemos hacer manualmente pulsando el botón *Add*.

Si tenemos los puntos suficientes, podemos arrancar el proceso de machaqueo de números pulsando *Optimizer* (figura 4). Comprobamos la opción *Positions (pairwise optim., starting from Anchor)* para el *Optimizer* y luego pulsamos *Optimize now!*. Las tres cajas de texto debajo muestran el número de la imagen y el valor que

Hugin necesita para corregir la imagen individual a continuación de los paréntesis cuadrados con o sin marca. Si una imagen no está marcada significa que Hugin no la modificará como ocurre con la imagen de referencia (Anchor). Si seleccionamos *Custom parameters below* para el *Optimizer* podemos ajustar manualmente los valores para cada imagen (esto es un ajuste para expertos).

La barra *Preview* en la parte superior nos muestra una imagen previa con los valores calculados. Puede que queramos seleccionar *Auto* en la ventana de vista previa para indicar al programa que nos muestre los últimos resultados.

Montando imágenes

La última pestaña, *Stitcher* (Coser), es el corazón del programa y el principal desafío al mismo tiempo: las cosas no siempre funcionan de la forma que se esperan. Coser es el proceso de poner múltiples imágenes juntas para crear una imagen panorámica.

Este proceso normalmente implica primero alinear las imágenes y corregir la distorsión antes de proceder al cosido. Hugin soporta los llamados motores de

costura: los motores de costura son módulos de software que realmente ajustan las imágenes panorámicas. El “costurero” Nona suministrado con el programa funciona muy bien, si bien la calidad no es extremadamente buena. Una cicatriz aparece en la mayoría de los casos.

Antes de que comencemos a coser debemos asegurarnos de que seleccionamos *Calculate Field of View* para recalcular el área de visión. Sólo el autor del programa conoce porque Hugin no hace esto automáticamente. También necesitamos modificar el tamaño de la imagen seleccionando *Calculate Optimal Size*. Si usamos el motor de costura Noma suministrado con el programa el formato de imagen preseleccionado es *JPG*, por lo que solo deberemos pulsar *Stitch now!*.

Transcurrido un poco de tiempo (o puede que bastante, dependiendo del tamaño y número de imágenes) tendremos de la imagen final en nuestro disco duro con el nombre que hemos seleccionado.

La alternativa es usar el programa externo *Enblend*, no soportado directamente por Hugin, lo que implica un paso más. Primero usamos Hugin para crear una serie de imágenes TIFF que estén correctamente alineadas pero no cosidas. Entonces seleccionamos la opción *into a layered TIFF file for Stitch the images* (ver figura 6).

El programa entonces selecciona automáticamente *Multiple TIFF* como el

Tabla 1: Lentes y campos de visión

Distancia focal	Campo de visión Panorama	Campo de visión Retrato
28mm	65	46
35mm	54	38
50mm	40	27

Caja 3: Mono para Autopano

Autopano-SIFT está escrito en el lenguaje de programación C# y necesita las librerías del proyecto Mono [3]. Para usar el software no necesitamos el entorno completo de Mono. Si necesitamos el interprete, el cual ejecuta el programa, y unas pocas librerías. Los paquetes de Suse son *mono-core-1.0.2-1.ximian.g.1.i586.rpm*, *libcuz6-2.6.2-1.ximian.g.o.i586.rpm*, *gtk-sharp-1.0.2-1.ximian.g.2.i586.rpm*, *libgtkhtml3.1_7-3.1.13.0.200405120525-0.s.nap.ximian.g.1.i586.rpm*. Los usuarios de Fedora no necesitan el paquete *libgtkhtml*.

Los usuarios de Debian deberían añadir el deposito Apt a */etc/apt/sources.list* e instalar los paquetes *mono*, *libgtk-cil* y *libglib-cil* a través de la línea de comandos. La página Web de Mono en Debian [4] nos ofrece más detalles.

Los archivos ejecutables escritos en Mono son lanzados escribiendo *mono programa.exe*. El módulo basado en el kernel Binfmt-Misc puede ejecutar el código Mono directamente. La mayoría de las distribuciones incluyen el módulo por defecto. Si nuestro montador automático no carga los sistemas de archivos requeridos para usar esta función debemos intentar el siguiente comando:

```
mount -t binfmt_misc none /proc/
sys/fs/binfmt_misc
```

Por último debemos indicarle al kernel como reconocer los archivos Mono y lanzar *mono*.

```
echo
```

```
' :CLR:M::MZ::/usr/bin/mono: '
> /proc/sys/fs/binfmt_misc/register
```

Suponiendo que nuestros archivos exe son ejecutables (*chmod +x*) ahora los podemos arrancar. Aún necesitamos desempaquetar Autopano-SIFT [5]. Para hacer esto podemos o arrancarlo directamente con */path/to/autopano-sift-2.0/bin/autopano.exe* o usar el programa Mono *mono/path/to/autopano-sift-2.0/bin/autopano.exe*. Si todo esto funciona, Autopano debe producir un corto texto de ayuda. Si *bin* está en nuestra variable ejecutable *PATH* (ruta) no necesitamos introducir la ruta completa. Está es también la mejor forma de comprobar la versión basada en GUI *autopanoog.exe*.

Para permitir que el código *autopano-complete.sh* funcione debemos ajustar la variable *PATH* como hemos descrito con anterioridad. De lo contrario el código mostrará un error indicando que no lo encontró.

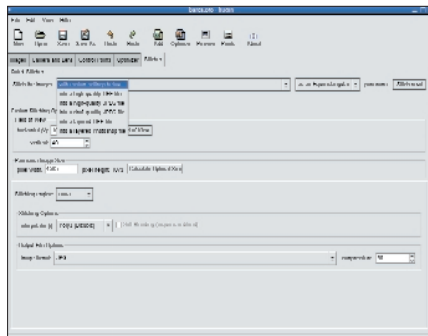


Figura 6: El módulo de costura de Hugin soporta varios formatos..

formato de salida. Pulsamos en *Stitch Now!*, y Hugin nos solicitará un nombre de archivo adjuntando un número de cuatro dígitos a los nombres de archivo de cada imagen. Entonces podemos lanzar una ventana terminal para coser las imágenes TIFF usando Enblend; usaremos *-o* para especificar el nombre del archivo:

```
enblend -o panorama.tif
pic0000.tif pic0001.tif
pic0002.tif
```

Los resultados son impresionantes. Enblend cose las imágenes sin feas cicatrices.

Automatizando puntos de control

Si todo esto implica demasiadas pulsaciones para nuestro gusto podemos usar otro programa que ajusta automática-

Cuadro 4: Enblend

El paquete Enblend [6] no necesita nada especial en la línea de librerías aparte de LibTIFF y LibJPG. En Fedora y Suse podemos instalar Enblend escribiendo

```
rpm -iv enblend-2.0-1.bp.fc2.
i386.rpm.
```

Desafortunadamente, los binarios no están disponibles para cada distribución y puede que necesitemos construir Enblend nosotros mismos. La siguiente línea se ocupa de la instalación en Debian:

```
apt-get install libjpeg62-dev
libtiff4-dev libpng3-dev fftw-
dev libboost-graph-dev lib
boost-dev libboost-python-dev
Ahora podemos proceder a compilar Enblend.
```

mente los puntos de control. Autopano-SIFT incluso busca emparejamientos que son invisibles al ojo humano. Para lanzar la herramienta ejecutamos el código no olvidando el nombre del archivo (usando el parámetro *-o*), analizado gramaticalmente luego por Hugin al igual que las imágenes individuales.

```
autopano-complete.sh -o output
left center right.jpg
```

como alternativa puede que prefiramos usar *autopanoog.exe*, basado en GUI, el cual hace exactamente lo mismo pero nos ahorra escribir el nombre de los archivos. Cuando cargamos el archivo PTO resultante con Hugin deberemos ajustar *degrees of view* en la pestaña *Camera and Lens*. Los pasos *Optimizer* y *Stitcher* son como se han descrito anteriormente.

¿Qué es lo siguiente?

Estas pistas nos deberían ayudar a comenzar a crear nuestras imágenes panorámicas propias. El hecho de que ni siquiera hemos visto una mínima parte del ingente número de opciones y menús disponibles muestra el potencial de herramientas como Hugin.

Una lista de correo amistosa nos ayudará a descubrir los últimos avances en el software panorámico, con usuarios muy felices de asistir a los desarrolladores. Es lógico pensar que los pequeños errores serán eliminados del programa en breve. Y si deseamos ayudar después de leer este artículo, ¿por qué no hacerlo?

RECURSOS

- [1] Nueva página Web para las huérfanas Panotools: <http://panotools.sourceforge.net/>
- [2] Hugin: <http://hugin.sourceforge.net/>
- [3] Descargas Mono: <http://www.mono-project.com/downloads/>
- [4] Mono en Debian: <http://pkg-mono.alioth.debian.org/>
- [5] Autopano-SIFT: <http://user.cs.tu-berlin.de/~nowozin/autopano-sift/>
- [6] Enblend: <http://enblend.sourceforge.net/>
- [7] Paquetes para varias herramientas de panorámicas: <http://bugbear.blackfish.org.uk/~bruno/panorama-tools/>