xvidcap

Airtime for Programs

At times, it can be quite difficult to describe an application's functions in words. A screenshot can help. If you need more than just a still picture, xvidcap can provide the answer. Now it is possible to make teaching aids that use recorded animations of you actually using the software so others can see exactly what you are doing.

BY ANDREA MÜLLER

Recording program steps from the desktop for demonstration or training purposes has never been an easy task. Although several dozen programs compete for the favor of users that produce documentation, none of them is a real movie mogul. Enter *xvidcap* [1] by Rasca Gmelch. The program, which is now maintained and developed by Karl H. Beckers, is tailor-made for creating on-screen demos. It can create images or MPEG files of user-definable desktop areas, and is thus perfectly suited to pointing the way through nested menu structures.

The Director's New Tools

Before you take your first steps as an on-screen movie director, you will obviously need to install the program. This is easy enough for users whose distributions support the RPM or Debian packages from the *xvidcap* homepage. The RPM provided by the program developer supports Suse Linux 9.0 and Red Hat 9.0, among others. There are two packages for Debian: one for the stable version

(*Woody*), and another for Debian Testing, a.k.a *Sarge*.

Users of other distros will need to compile *xvidcap* themselves. This assumes that you have installed the developer packages for *libpng*, *libjpeg*, and *XFree86* on your system. Just follow the standard procedure: *./configure ; make ; su -c make install* to build *xvidcap* and install the program in the directory structure below */usr/local*. As *root* is the only user with write permissions for this level, you may need to enter *su -c* to temporarily assume *root* privileges for the *make install* step.

xvidcap relies on the *ffmpeg* [2] program to create video clips. If this is not installed on your system, your version of

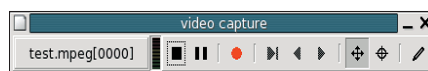**Figure 1: The desktop movie creator xvidcap.**

Figure 2: The gtk-2 variant has a more modern look than the original.

xvidcap will still be able to create screenshots, but will not handle videos. If your distribution does not include the *ffmpeg* package, you will need to build the *ffmpeg* version included with the *xvidcap* distribution. To do so, add the *—with-forced-embedded-ffmpeg* parameter to the *./configure* command to create an *xvidcap* with *embedded ffmpeg* support, or change to the *ffmpeg* subdirectory before compiling *xvidcap*. Enter *./configure ; make ; su -c make install* in this directory to create the *ffmpeg* encoder, and the *ffplay* player, and then type *make install* to copy these tools to */usr/local/bin*. The advantage of this approach is that it allows you to use *ffmpeg* and *ffplay* independently of *xvidcap*, which the embedded variant does not allow you to do.

The *—with-gtk2 ./configure* option is also interesting if you are building the program yourself, as it creates a program called *gvidcap*. In contrast to the original, *gvidcap* has a modern **gtk-2** GUI which perfectly matches the current Gnome desktop.
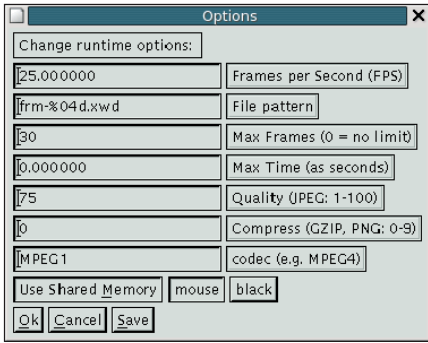
Figure 3: Use this dialog to specify the output format and the output file name.

The Script

To launch the new tool installed in the last step, type

```
xvidcap &
```

or enter

```
gvidcap &
```

to launch the *gtk-2* version. The control elements for *xvidcap* (see Figure 1) and *gvidcap* (see Figure 2) are organized in exactly the same way. This has an upside and a downside. The original version may look dull, but it is certainly more stable and more highly configurable than the more modern version. Stability, configurability, and the fact that both interfaces are similar, are what made us decide to concentrate on *xvidcap* in this article.

When you launch the program, it displays a red frame about the size of a credit card to indicate the recording area. The frame moves with the program window. To prevent this, you can click on the x button to detach the frame from the rest of the application. As the default size is too small for anything but the tiniest of tools, you will want to define the desktop area that you are interested in recording before you do anything else. Click on the cross-hair in the toolbar to
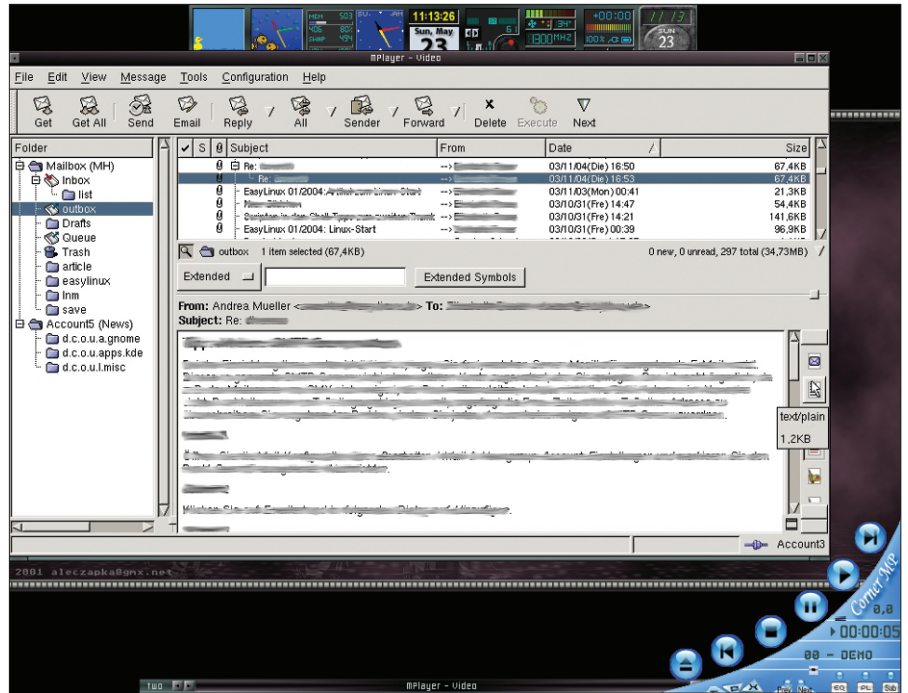

Figure 4: MPlayer playing a video about email readers created with xvidcap.

change the mouse cursor to a cross-hair. You can then either hold down the left mouse button and drag the mouse to define an area, or click on a window, assuming that you are only interested in this window's content. If you select a window, the video clip will not include that window's title bar.

After selecting your location, the next step is to define a few parameters, such as the output format. [Alt-O] (or [Ctrl-P] for *gvidcap*) will open the configuration dialog shown in Figure 3.

First of all, set the value for *Max Frames* to 0, as *xvidcap* has a default restriction of 30 images. A value of 0 removes this limit, allowing you to create video clips of any length. Use the *Frames per Second (FPS)* box to specify the number of images per second *xvidcap* will capture. The default is 25 – don't be confused by the way this is displayed (*25.000000*). If you need to create **MPEG**1 or MPEG4 video clips, check out

the *xvidcap* manpage (option —*fps*) for the required frames per second values. For example, *25* and *50* are valid frame rates for MPEG1, whereas MPEG4 can use values between *7.5* and *30*. If you enter an invalid value, your video clips will either run too fast, or too slow.

Enter the video codec *(e.g. MPEG4)* into the *codec* field. This defaults to *MPEG1*, but you may want to change this to *MPEG4* for enhanced recording quality. The resolution of the MPEG1 videos we created was so poor that we were unable to decipher some of the menus at times.

Video or Single Image?

You still need to type the output file name into the *File pattern* box. If you leave the default, *frm-%04d.xwd*, *xvidcap* will not create a video, but instead generate single *xwd* (*x w*indow *dump*) formatted files in the directory where you launched the program. The *%04d* string adds a serial number to the image file name, creating the files *frm-0000.xwd* through *frm-9999.xwd*. The percent sign indicates formatting information: *04d* (*d* for decimal) specifies the number of digits. If your demo will have more than 10,000 images, simply change the value for *File pattern* to *frm-%05d.xwd*, or even *frm-%06d.xwd*. You can modify both the base file name (*frm*)

GLOSSARY

gtk-2:: *The Gimp Toolkit with the major version 2. This graphical library provides developers with ready-made widgets for GUI-based applications, such as radio buttons, dialog boxes, and input fields. Besides the Gnome desktop version 2.0 or later, the new Gimp is also based on gtk-2.*

MPEG: *Abbreviation for Motion Pictures Expert Group. This group, which comprises enterprises and universities, standardized compression techniques for video formats. MPEG encoding attempts to describe as many image segments as possible by specifying how they differ from the previous image; this in turn allows for smaller file sizes.*

and the output format. *xvidcap* can also create JPEG or PNG files. Single images are useful if you need to edit the results or if you are trying to get your hands on a screenshot that programs like *ksnapshot* can not deliver, such as an error message that pops up on your screen and immediately disappears.

To create a video directly, change the entry in *File pattern* to any file name using *mpeg* as the extension, for example *programdemo.mpeg*. To avoid modifying the options each time you use the program, click *Save*. This tells *xvidcap* to store your settings in the *.xvidcap.scf* file below your home directory. The *gtk-2* counterpart, *gvidcap*, reads its defaults from the same file. In other words, any changes you make in one program will immediately take effect in the other. Deleting ~/.xvidcap.scf will restore the program defaults.

Take One, Action!

Click the button with the red dot to start recording the desktop area within the red frame. After demonstrating the software aspects you are interested in, click on the button with the black square to stop recording. You can use a player like MPlayer (see Figure 4) to play back your demo to an appreciative audience. To keep things simple, you can opt to use the *ffplay* command-line player supplied with *ffmpeg*. The following command launches the tool

```
ffplay videodemo.mpeg
```

Pressing [q] quits the show.

After creating a series of images, you can click on the buttons with the left or right arrows to flip through the individual images. Clicking on the button
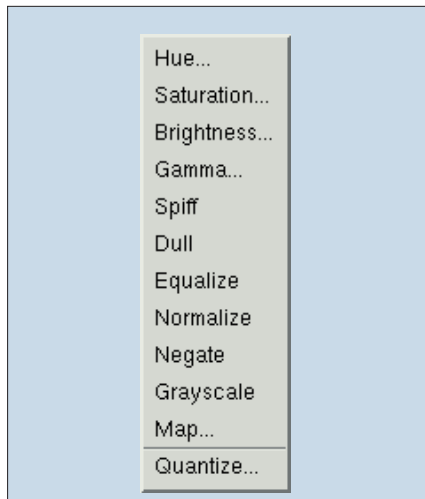
Figure 5: xvidcap uses Display to open individual images. Display has a number of image manipulation features. The Enhance menu has options for modifying the brightness, gamma values and saturation.

with the pencil image on the right opens the current image in *display*, the image viewer from the **ImageMagick** package. You can then click on the image window to pop up a menu with a number of image manipulation functions (see Figure 5) such as color (*Enhance*) and size manipulation, or rotation (*Transform*) features, and a number of *Effects* (*F/X*).

The button with the filmstrip calls *animate*, another tool from the ImageMagick collection. As the name suggests, *animate* creates and plays an animation made up of the individual image files. Avoid using this option with collections of more than 400 images. Feeding too much material to *animate* will bring even the most powerful system to its knees.

From Single Images to Video Clips

File | Make Video converts your images to a video clip. *xvidcap* parses the ~/.Xdefaults to discover what program it should use for this task. This is where the defaults for the graphical programs are stored. This said, *gvidcap* does not honor the assignments in ~/.Xdefaults, so the conversion command will only apply to *xvidcap*. Let's assume that you have a collection of PNG-formatted images, with names based on a format string such as *pic%04d.png*, and that you want to use *ffmpeg* to convert these

images to a MPEG1 video clip. The following line

```
xvidcap*mkVideoCommand:ffmpeg ⏎
-i %s video.mpeg
```

in ~/.Xdefaults tells *xvidcap* to call *ffmpeg* when you select *Make Video*. For this to work, you need the *ffmpeg* program proper: you can not use the *xvidcap* embedded variant. *xvidcap* replaces the *%s* string with the name pattern used by the individual image files – this was *pic%04d.png* in our example. As *ffmpeg* recognizes this naming convention, it can convert files with names such as *pic0000.png*, *pic0001.png* one by one and store the result in the output file, *video.mpeg*, below the current directory.

There is a pitfall to watch out for when using *ffmpeg* to generate a video from individual image files. The encoder can not handle the *xwd* files that *xvidcap* creates by default. *ffmpeg* uses the following graphics formats: *pnm*, *pam*, *pgmyuv*, *yuv*, *png*, *jpeg*, and *gif*. The PNG format is a good choice when grabbing screenshots with *xvidcap*. To select the PNG format, select the PNG file extension as the *File pattern* in the *xvidcap* options menu.

The quickest way to success for amateur screen directors is to tell *xvidcap* to create video output, and leave individual image processing to users who are unsatisfied with "run-of-the-mill" video clips, and want to modify the raw material before generating a video clip. ■

INFO

[1] xvidcap: *http://sourceforge.net/projects/xvidcap*

[2] ffmpeg: *http://ffmpeg.sourceforge.net/*

THE AUTHOR

After nearly two years as an independent journalist, Andrea Mueller has now taken up a new editorial position at Linux New Media AG.

When she is not taking care of articles, or wrapping packages for the EasyLinux download area, Andrea looks beyond Linux and is involved with other operating systems such as QNX, BeOS and NetBSD.

GLOSSARY

ImageMagick: *A collection of ten image manipulation tools. As most of these tools are command line based, ImageMagick is the perfect choice for scripting image conversion and manipulation jobs.*

~/.Xdefaults: *This is where users can define preferences for GUI-based programs, such as the background color or the font. To find out which preferences an application can use, you need to check the manpage for that application. Current KDE and Gnome applications typically ignore the settings in ~/.Xdefaults.*