Inessential Bitmaps (XBM format)

The Student Information Processing Board

Eri Izawa

March 7, 1994

1 Introduction: What are Bitmaps?

Welcome to bitmapping! If you have ever wanted to draw or design your own screen backgrounds, draw up your own xscreensaver pattern, make artwork for your World Wide Web homepage, or just plain try out your artistic urges on Athena, bitmaps may well be just what you wanted. This guide details how to create and display XBM format bitmaps, which are black and white and easily displayable on your background screen or on the WWW.

A bitmap is basically a map of bits (surprise, surprise) — but when taken altogether and viewed through bitmap programs, the patterns of bits form a picture. Sometimes the pictures can be pretty impressive. If you have had any experience with painting programs (such as Mac Paint, Super Paint, or such), then you will find the format reassuringly similar.

One note about large bitmaps: they take up a lot of space in your account. It's a good idea to keep your large bitmaps compressed (type compress <filename> to compress, uncompress <filename> to uncompress) to save space when you aren't working on them. Luckily, there are many ways to view even compressed bitmap files.

2 Making Bitmaps

Making xbm-format bitmaps is just like making your common paint-program generated picture. If you've never used a paint program before, don't worry; it only takes a minute to learn — although, as with any art, the more you practice, the more you can do with it.

2.1 Where the Programs Live

Many different bitmap programs exist here and there. I suggest the following:

```
% attach x11r5
% /mit/x11r5/$bindir/bitmap
```

(The `afs` path is: `/afs/athena/astaff/project/x11r5/$bindir/bitmap`.)

Note that this program for some reason is not capable of loading large (about 300 by 300 and larger) bitmaps, though it is perfectly usable for making them. Hopefully this bug will be corrected soon. Another bug it seems to have is that enlarging the edit window *after* a bitmap is loaded may cause the program to become unusable. To avoid this, start up the program, enlarge it, and then load your bitmap. Aside from these bugs, however, it is a generally good program.

Another all around good program is xpaint. First `attach graphics`, and then do:

`% /mit/graphics/$bindir/xpaint`

(The `afs` path is: `/afs/athena/contrib/graphics/bin/$bindir/xpaint`)

This is a color painting program; however, since bitmaps do not store color information, the colors are meaningless and you should stick to black and white. The program is slightly flaky. However, it has such useful features as Zoom, although in a different format than standard bitmap programs (you need to go to the Image menu and select the Change Zoom feature, and from there type in something like '2' or '3' instead of '1'; then use the scroll bars to move around). Xpaint is also useful for converting bitmaps to other formats — but this will be discussed later.

Do not confuse xpaint with dxpaint, a Decstation-only bitmap-making program that seems to have at least one major bug.

Another program that creates bitmap is in sipb. `add sipb`, and then type `xfig &` to start the program. It is *not* a standard bitmap editor (it is a drawing program, not a paint program) — in fact, it can not edit bitmap files. It can, however, save its drawings in bitmap format (choose option `Export`, and from there choose, as your format, `x11 bitmap`).

I do not recommend the default bitmap program in `/usr/athena/bin/bitmap`, though you may have to if none of the above work. It lacks several important features, like Zoom and Undo, but it tends to work on most machines, and should be fine for small bitmaps.

2.2 Using the program

The basic idea with most paint and bitmap programs is pretty obvious — point at the square you want colored in and click there. Most programs are also self-obvious when it comes to their various features. Play around and experiment if you're not sure what something does. Don't forget to save your bitmap in your home directory. Some people end the bitmap names with `.xbm`, but this is not necessary.

Some techniques you might want to try:

FLOOD FILL: This is wonderful for turning areas white or black (usually the left mouse key for black, the right mouse key for white). Be very careful with FLOOD FILL, lest you wind up with a completely black bitmap by accident; make sure areas you want to FLOOD FILL are completely bounded by a solid line. Always be ready to UNDO after a FLOOD FILL. Bitmap programs that lack UNDO should be avoided for this reason.

ZOOM: If you're working on a large (bigger than 100x100) bitmap, use the ZOOM feature to do detail work and touch-ups. This is probably the most important feature after FLOOD FILL.

PATTERN FILL: To create something that looks like medium grey on a b/w monitor, fill in the grid in a checkerboard fashion, alternating black with white (use the ZOOM if necessary). Doing this with the LINE feature is often easier than going dot-by-dot. To make darker or lighter greys, add more black or white squares in a consistent, regular fashion. You may also make any number of textures or patterns like this. To fill in large regions of a bitmap with the pattern, I suggest selecting, copying, and pasting pieces of the pattern in appropriate places (being careful so that the pattern meshes smoothly with other sections that contain the same pattern). If you ever find a bitmap program that will FLOOD FILL an area with a specific pattern, by all means use that instead.

FILLED Shape: With the `x11r5` bitmap program, using the FILLED RECTANGLE or CIRCLE feature while depressing the left mouse button will create a black shape; depressing the right button will create a white shape, and the middle will create a shape that contains whatever color contrasts with the background.

3 Showing Bitmaps — Basic Stuff

If you just want to put up one copy of the bitmap and be able to click it away, use `showbitmap`. You can `add consult` and then do `showbitmap -rv <bitmap name>`.

(The path is `/mit/consult/$bindir/showbitmap`.)

The -rv flag merely inverts the bitmap coloration; for some reason, showbitmap puts up bitmaps inverted on some machines, and thus the -rv is necessary to see the original form.

Another way to put up one copy of a bitmaps, even compressed ones, is to use:

```
% /mit/graphics/$bindir/xloadimage <bitmap name>
```

To put a bitmap up on your root screen (the background), use:

```
% xsetroot -bitmap <bitmap name>
```

This will place a tile pattern of your bitmaps onto your root screen. If you're using compressed bitmaps, you can use the following to do the same thing:

```
% /mit/graphics/$bindir/xloadimage -onroot <bitmap name>
```

4 Showing Bitmaps — Background Slideshow

If you're like me and have a bunch of different bitmaps that you like, but hate to manually change your background, here's a nifty little program that will do it for you. Just type it up into an emacs buffer, save it (I call mine XSRB1 and XSRB.un), and then run it as a background process (just type `XSRB1 &` at the prompt). To stop a backgrounded process, do `ps -ux` and then type `kill -9 <PID number>`.

4.1 "XSRB1"

(Note: this works for compressed bitmaps, but you must have graphics added or in your path. To add graphics, merely type `add graphics`. This will also work for non-compressed bitmaps.)

```
#!/bin/csh -f
set pictures = `ls /mit/rei/bitart/Backgrounds`
set num = 1
while ($num)
        foreach i ($pictures)
                xloadimage -onroot /mit/rei/bitart/Backgrounds/$i >& /dev/null
                sleep 60
        end
end
```

4.2 "XSRB.un"

(Note: this works only on uncompressed bitmaps.)

```
#!/bin/csh -f
set pictures = `ls /mit/rei/bitart/IS`
set num = 1
while ($num)
        foreach i ($pictures)
                xsetroot -bitmap /mit/rei/bitart/IS/$i >& /dev/null
        end
end
```

Note: the above program does not have a sleep time because it was designed to produce an "animation" by throwing up slightly different bitmaps one after the other.

5 Putting Bitmaps into your World Wide Web homepage

If you use the World Wide Web, you know that it can put up some really nice images, including bitmaps and compressed bitmaps.

In your homepage file (`<filename>.html`), put in something like the following:

```
<IMG SRC="http://www.mit.edu:8001/afs/athena.mit.edu/user/r/e/rei/bitart/drae">
```

This will insert that bitmap. If your file is compressed, add `.Z` to the end of the name. If the bitmap is especially large, you might also want to move it out of your homepage.

6 Putting Bitmaps in Startup Files

To get a specific bitmap on your root screen whenever you login, put something like this in your .startup.X file:

```
xloadimage -onroot ~/bitart/Backgrounds/Respite &
/mit/sipb/$bindir/xscreensaver -geometry -25-55
```

If you want a screensaver to start up, you should also add:

```
/mit/sipb/$bindir/xscreensaver -geometry -25-55
```

Of course, substitute your own bitmap and pathname in! Then, to customize your screensaver icon, put something like the below in your .Xresources file:

```
xscreensaver*icon.bitmap:/mit/rei/bitart/drae
xscreensaver*float.bitmap:/mit/rei/bitart/dolphin
xscreensaver*useBackground:true
```

The `float.bitmap` section tells xscreensaver to use a different bitmap for when the icon floats around on the screen.

7 Printing Bitmaps

If you create a bitmap and want to take a hardcopy home to tape on your door, you can print it like this:

```
% add consult
% showbitmap <bitmapname> &
% xdpr -P<printername>
```

and click on the window with the bitmap once your pointer turns into a crosshatch. Xdpr tries to print bitmaps as large as possible. If you want your little bitmap to look like a little bitmap when printed, do:

```
% xdpr -scale <number> -P<printername>
```

where `<number>` scales the image's size up by that factor. `1` is absolutely tiny. `4` is about right if you want the paper image to be about the same size as the screen image.

Another way of printing out bitmaps is listed below, and involves converting the bitmap into PostScript.

8 Advanced Stuff

8.1 Converting to PostScript and Inserting into Files

The graphics locker contains several programs to convert bitmaps to other formats such as PostScript (which can be inserted into LaTeX documents).

Both xpaint and xv have this capability. Basically, load (or open) your xbm format bitmap through either program, and then save it as a PostScript file (add `.ps` to the end of its name). Now, in PostScript form, it can be printed out or added to your documents. Both programs save files in Encapsulated PostScript form.

Then, in your LaTeX document, you need to add `epsf` to the `\documentstyle[]` command, and then put in something like the following:

```
\hspace*{1mm}
\vspace{2cm}
\epsfbox{filename.ps}
```

where you want the bitmap.

One last thing: you will have to `attach sipb` and then do

```
% setenv TEXINPUTS .:/mit/sipb/lib/tex/macros:/usr/athena/lib/tex/macros+
```

before trying to `latex` your document.

If you have problems, see the Inessential LaTeX guide for more details on this and other LaTeX tricks. In-depth information is also available in the More LaTeX on Athena: Reports and Advanced Topics publication, available at Graphic Arts. You may also access it via online `help`: in the main menu, go to the Archive of Online Documentation, and from there go to the Archive of Athena Documents Available at MIT Graphic Arts.

8.2 Sending Bitmaps to Other Terminals

If someone at another terminal wants to see your wonderful bitmaps, you may send them directly to that person's screen. First, the other person must xhost +<your machine's name>. Then, you may do the following:

```
% xsetroot -display <the other machine name>:0 -bitmap <bitmapname>
```

9 Where to Find Sample Bitmaps

You may use any of the programs mentioned above to go around and view other people's bitmaps; in addition, you may find some on the World Wide Web, in various homepages and elsewhere.

Places to find neat bitmaps include:

```
/mit/rei/bitart/
/mit/rei/bitart/Backgrounds/
/afs/athena/user/r/j/rjbarbal/Bitmaps
/usr/lib/X11/bitmaps/
/mit/sipb/lib/bitmap/
/mit/dcctdw/Bitart/
/mit/dcctdw/Bitart.more/
/mit/yandros/stuff/graphics/
```

Note: to access some of the above, you may need to type aklog sipb first. Also, some of the bitmaps are in .gz format, a special compression format. You may need to copy them and unzip them (gzip and gunzip are both available in the sipb locker) to view them yourself. Anything ending in just .Z is probably a bitmap that you should be able to view with xloadimage, as outlined above. Also, remember that .xbm is sometimes used to denote xbm format bitmaps.

You may also use WWW (type Mosaic & at your prompt) and find rei's homepage from the SIPB homepage, where I have some of my bitmaps available for viewing.

Hopefully some of the examples will give you a general idea of what is possible with xbm-format bitmaps. The rest is up to you.

If you have questions about anything in this document, or find new bitmap programs, feel free to email me at rei@mit.edu, or stop by SIPB.