Tifftopnm User Manual

tifftopnm

Updated: 27 March 2005
Table Of Contents

NAME

tifftopnm - convert a TIFF file into a PNM image

SYNOPSIS

tifftopnm [**-alphaout=**{*alpha-filename*,**-**}] [**-headerdump**]
[**-respectfillorder**] [**-byrow**] [*tiff-filename*]

You may abbreviate any option to its shortest unique prefix. You may use two hyphens instead of one in options. You may separate an option and its value either by an equals sign or white space.

DESCRIPTION

This program is part of Netpbm.

tifftopnm reads a TIFF file as input and produces a PNM image as output. The type of the output file depends on the input file - if it's black & white, generates a PBM image; if it's grayscale, generates a PGM image; otherwise, a PPM image. The program tells you which type it is writing.

If the TIFF file contains multiple images (multiple "directories,"), **tifftopnm** generates a multi-image PNM output stream. Before Netpbm 10.27 (March 2005), however, it would just ignore all but the first input image.

This program cannot read every possible TIFF file -- there are myriad variations of the TIFF format. However, it does understand monochrome and gray scale, RGB, RGBA (red/green/blue with alpha channel), CMYK (Cyan-Magenta-Yellow-Black ink color separation), and color palette TIFF files. An RGB file can have either single plane (interleaved) color or multiple plane format. The program reads 1-8 and 16 bit-per-sample input, the latter in either bigendian or littlendian encoding. Tiff directory information may also be either bigendian or littlendian.

There are many TIFF formats that **tifftopnm** can read only if the image is small enough to fit in memory. **tifftopnm** uses the TIFF library's TIFFRGBAImageGet() function to process the TIFF image if it can get enough memory for TIFFRGBAImageGet() to store the whole image in memory at once (that's what TIFFRGBAImageGet() does). If not, **tifftopnm** uses a more primitive row-by-row conversion strategy using the raw data returned by TIFFReadScanLine() and native intelligence. That native intelligence does not know as many formats as TIFFRGBAImageGet() does. And certain compressed formats simply cannot be read with TIFFReadScanLine().

Before Netpbm 10.11 (October 2002), **tifftopnm** never used TIFFRGBAImageGet(), so it could not interpret many of the formats it can interpret today.

There is no fundamental reason that this program could not read other kinds of TIFF files even when they don't fit in memory all at once. The existing limitations are mainly because no one has asked for more.

The PNM output has the same maxval as the Tiff input, except that if the Tiff input is colormapped (which implies a maxval of 65535) the PNM output has a maxval of 255. Though this may result in lost information, such input images hardly ever actually have more color resolution than a maxval of 255 provides and people often cannot deal with PNM files that have maxval > 255. By contrast, a non-colormapped Tiff

image that doesn't need a maxval > 255 doesn't have a maxval > 255, so when **tifftopnm** sees a non-colormapped maxval > 255, it takes it seriously and produces a matching output maxval.

Another exception is where the TIFF maxval is greater than 65535, which is the maximum allowed by the Netpbm formats. In that case, **tifftopnm** uses a maxval of 65535, and you lose some information in the conversion.

The *tiff-filename* argument names the regular file that contains the Tiff image. If you specify "-" or don't specify this argument, **tfftopnm** uses Standard Input. In either case, the file must be seekable. That means no pipe, but any regular file is fine.

OPTIONS

-alphaout=*alpha-filename***tifftopnm** creates a PGM file containing the alpha channel values in the input image. If the input image doesn't contain an alpha channel, the *alpha-filename* file contains all zero (transparent) alpha values. If you don't specify **-alphaout**, **tifftopnm** does not generate an alpha file, and if the input image has an alpha channel, **tifftopnm** simply discards it.

If you specify **-** as the filename, **tifftopnm** writes the alpha output to Standard Output and discards the image.

See **pamcomp** for one way to use the alpha output file.

-respectfillorder By default, **tifftopnm** ignores the "fillorder" tag in the TIFF input, which means it may incorrectly interpret the image. To make it follow the spec, use this option. For a lengthy but engaging discussion of why **tifftopnm** works this way and how to use the **-respectfillorder** option, see the note on fillorder below.

-byrow This option can make **tifftopnm** run faster.

tifftopnm has two different ways to do the conversion from Tiff to PNM, using two different facilities of the TIFF library:

Whole ImageDecode the entire image into memory at once, using TIFFRGBAImageGet(), then convert to PNM and output row by row.

Row By RowRead, convert, and output one row at a time using TIFFReadScanline().

Whole Image is preferable because the Tiff library does more of the work, which means it understands more of the Tiff format possibilities now and in the future. Also, some compressed TIFF formats don't allow you to extract an individual row.

Row By Row uses far less memory, which means with large images, it can run in environments where Whole Image cannot and may also run faster. And because Netpbm code does more of the work, it's possible that it can be more flexible or at least give better diagnostic information if there's something wrong with the TIFF.

In Netpbm, we stress function over performance, so by default we try Whole Image first, and if we can't get enough memory for the decoded image or TIFFRGBAImageGet() fails, we fall back to Row By Row. But if you specify the **-byrow** option, **tifftopnm** will not attempt Whole Image. If Row By Row does not work, it simply fails.

See Color Separation (CMYK) TIFFs for a description of one way Row By Row makes a significant difference in your results.

Whole Image costs you precision when your TIFF image uses more than 8 bits per sample. TIFFRGBAImageGet() converts the samples to 8 bits. **tifftopnm** then scales them back to maxval 65535, but the lower 8 bits of information is gone.

Before Netpbm 10.11 (October 2002), **tifftopnm** always did Row By Row. Netpbm 10.12 always tried Whole Image first. **-byrow** came in with Netpbm 10.13 (January 2003).

-headerdump Dump TIFF file information to stderr. This information may be useful in debugging TIFF file conversion problems.

NOTES

Fillorder

There is a piece of information in the header of a TIFF image called "fillorder." The TIFF specification quite clearly states that this value tells the order in which bits are arranged in a byte in the description of the image's pixels. There are two options, assuming that the image has a format where more than one pixel can be represented by a single byte: 1) the byte is filled from most significant bit to least significant bit going left to right in the image; and 2) the opposite.

However, there is confusion in the world as to the meaning of fillorder. Evidence shows that some people believe it has to do with byte order when a single value is represented by two bytes.

These people cause TIFF images to be created that, while they use a MSB-to-LSB fillorder, have a fillorder tag that says they used LSB-to-MSB. A program that properly interprets a TIFF image will not end up with the image that the author intended in this case.

For a long time, **tifftopnm** did not understand fillorder itself and assumed the fillorder was MSB-to-LSB regardless of the fillorder tag in the TIFF header. And as far as I know, there is no legitimate reason to use a fillorder other than MSB-to-LSB. So users of **tifftopnm** were happily using those TIFF images that had incorrect fillorder tags.

So that those users can continue to be happy, **tifftopnm** today continues to ignore the fillorder tag unless you tell it not to. (It does, however, warn you when the fillorder tag does not say MSB-to-LSB that the tag is being ignored).

If for some reason you have a TIFF image that actually has LSB-to-MSB fillorder, and its fillorder tag correctly indicates that, you must use the **-respectfillorder** option on **tifftopnm** to get proper results.

Examples of incorrect TIFF images are at ftp://weather.noaa.gov. They are apparently created by a program called **faxtotiff**.

This note was written on January 1, 2002.

Color Separation (CMYK) TIFFs

Some TIFF images contain color information in CMYK form, whereas PNM images use RGB. There are various formulas for converting between these two forms, and **tifftopnm** can use either of two.

The TIFF library (Version 3.5.4 from libtiff.org) uses Y=(1-K)*(1-B) (similar for R and G) in its TIFFRGBAImageGet() service. When **tifftopnm** works in Whole Image mode, it uses that service, so that's the conversion you get.

But when **tifftopnm** runs in Row By Row mode, it does not use TIFFRGBAImageGet(), and you get what appears to be more useful: Y=1-(B+K). This is the inverse of what **pnmtotiffcmyk** does.

See the **-byrow** option for more information on Whole Image versus Row By Row mode.

Before Netpbm 10.21 (March 2004), **tifftopnm** used the Y=(1-K)*(1-B) formula always.

SEE ALSO

pnmtotiff, **pnmtotiffcmyk**, **pamcomp**, **pnm**

AUTHOR

Derived by Jef Poskanzer from tif2ras.c, which is Copyright (c) 1990 by Sun Microsystems, Inc. Author: Patrick J. Naughton (naughton@wind.sun.com).

Table Of Contents

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
NOTES
Fillorder
Color Separation (CMYK) TIFFs
SEE ALSO
AUTHOR