Pnmtojbig User Manual

PNMTOJBIG

Updated: 20 May 2000
Table Of Contents

NAME

pnmtojbig - PNM to JBIG file converter

SYNOPSIS

pnmtojbig [*options*] [*input-file* [*output-file*]]

DESCRIPTION

This program is part of Netpbm.

pnmtojbig reads a PBM or PGM image, compresses it, and outputs the image as a JBIG bi-level image entity (BIE) file.

JBIG is a highly effective lossless compression algorithm for bi-level images (one bit per pixel), which is particularly suitable for scanned document pages.

A JBIG encoded image can be stored in several resolutions (progressive mode). These resolution layers can be stored all in one single BIE or they can be stored in several separate BIE files. All resolution layers except the lowest one are stored merely as differences to the next lower resolution layer, because this requires less space than encoding the full image completely every time. Each resolution layer has twice the number of horizontal and vertical pixels than the next lower layer. JBIG files can also store several bits per pixel as separate bitmap planes, and **pnmtojbig** can read a PGM file and transform it into a multi-bitplane BIE.

OPTIONS

-q Encode the image in one single resolution layer (sequential mode). This is usually the most efficient compression method. By default, the number of resolution layers is chosen automatically such that the lowest layer image is not larger than 640 × 480 pixels.

-x *number*Specify the maximal horizontal size of the lowest resolution layer. The default is 640 pixels.

-y *number*Specify the maximal vertical size of the lowest resolution layer. The default is 480 pixels.

-l *number*Select the lowest resolution layer that will be written to the BIE. It is possible to store the various resolution layers of a JBIG image in progressive mode into different BIEs. Options **-l** and **-h** allow you to select the resolution-layer interval that will appear in the created BIE. The lowest resolution layer has number 0 and this is also the default value. By default, **pnmtojbig** writes all layers.

-h *number*Select the highest resolution layer that will be written to the BIE. By default, **pnmtojbig** writes all layers. See also option **-l**.

-b Use binary values instead of Gray code words in order to encode pixel values in multiple bitplanes. This option has only an effect if the input is a PGM file and if more than one bitplane is produced. Note that the decoder has to make the same selection but cannot determine from

the BIE, whether Gray or binary code words were used by the encoder.

-d *number*Specify the total number of differential resolution layers into which the input image will be split in addition to the lowest layer. Each additional layer reduces the size of layer 0 by 50 %. This option overrides options **-x** and **-y**, which are usually a more comfortable way of selecting the number of resolution layers.

-s *number*The JBIG algorithm splits each image into a number of horizontal stripes. This option specifies that each stripe shall have *number* lines in layer 0. The default value is selected so that approximately 35 stripes will be used for the whole image.

-m *number*Select the maximum horizontal offset of the adaptive template pixel. The JBIG encoder uses a number of neighbour pixels in order to get statistical a priori knowledge of the probability, whether the next pixel will be black or white. One single pixel out of this template of context neighbor pixels can be moved around. Especially for dithered images it can be a significant advantage to have one neighbor pixel which has a distance large enough to cover the period of a dither function. By default, the adaptive template pixel can be moved up to 8 pixels away. This encoder supports up to 23 pixels, however as decoders are only required to support at least a distance of 16 pixels by the standard, no higher value than 16 for *number* is recommended in order to maintain interoperability with other JBIG implementations. The maximal vertical offset of the adaptive template pixel is always zero.

-t *number*Encode only the specified number of most significant bit planes. This option allows to reduce the depth of an input PGM file if not all bits per pixel are needed in the output.

-o *number*JBIG separates an image into several horizontal stripes, resolution layers and planes, were each plane contains one bit per pixel. One single stripe in one plane and layer is encoded as a data unit called stripe data entity (SDE) inside the BIE. There are 12 different possible orders in which the SDEs can be stored inside the BIE and *number* selects which one shall be used. The order of the SDEs is only relevant for applications that want to decode a JBIG file which has not yet completely arrived from e.g. a slow network connection. For instance some applications prefer that the outermost of the three loops (stripes, layers, planes) is over all layers so that all data of the lowest resolution layer is transmitted first.

The following values for *number* select these loop arrangements for writing the SDEs (outermost loop first):

0 planes, layers, stripes

2 layers, planes, stripes

3 layers, stripes, planes

4 stripes, planes, layers

5 planes, stripes, layers

6 stripes, layers, planes

All loops count starting with zero, however by adding 8 to the above order code, the layer loop can be reversed so that it counts down to zero and then higher resolution layers will be stored before lower layers. Default order is 3 which writes at first all planes of the first stripe and then completes layer 0 before continuing with the next layer and so on.

-p *number*This option allows you to activate or deactivate various optional algorithms defined in the JBIG standard. Just add the numbers of the following options which you want to activate in order to get the *number* value:

4 deterministic prediction (DPON)

8 typical prediction (TPBON)

16 diff. layer typical prediction (TPDON)

64 layer 0 two-line template (LRLTWO)

Except for special applications (like communication with JBIG subset implementations) and for debugging purposes you will normally not want to change anything here. The default is 28, which

provides the best compression result.

-c The adaptive template pixel movement is determined as suggested in annex C of the standard. By default the template change takes place directly in the next line which is most effective. However a few conformance test examples in the standard require the adaptive template change to be delayed until the first line of the next stripe. This option selects this special behavior, which is normally not required except in order to pass some conformance test suite.

-v After **pnmtojbig** creates the BIE, it lists a few technical details of the created file (verbose mode).

FORMATS

Most of the format **pnmtojbig** creates is defined by the JBIG standard.

The standard, however, does not specify which values in the BIE mean white and which mean black. It contains a recommendation that for a single plane image zero mean background and one mean foreground, but the Netpbm formats have no concept of foreground and background. And the standard says nothing about values for multiple plane BIEs.

pnmtojbig follows Markus Kuhn's implementation of the standard in the **pbmtojbg** program that comes with his JBIG library: If the BIE is a single plane BIE, zero means white and one means black. If it is a multiple plane BIE, zero means black and the maximal value is white.

STANDARDS

This program implements the JBIG image coding algorithm as specified in ISO/IEC 11544:1993 and ITU-T T.82(1993).

AUTHOR

pnmtojbig is based on the JBIG library by Markus Kuhn, part of his **JBIG-KIT** package. The **pbmtojbg** program is part of the *JBIG-KIT* package. The most recent version of that library and tools set is freely available on the Internet from anonymous ftp server ftp.informatik.uni-erlangen.de in directory pub/doc/ISO/JBIG/.

pnmtojbig is part of the Netpbm package of graphics tools.

SEE ALSO

pnm, **jbigtopnm**

LICENSE

If you use **pnmtojbig**, you are using various patents, particularly on its arithmetic encoding method, and in all probability you do not have a license from the patent owners to do so.

Table Of Contents

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
FORMATS
STANDARDS
AUTHOR
SEE ALSO
LICENSE