JBIG under Windows

Authored by

A. BAKALIDIS
Imaging Lab
Electrical Engineering Department
Democritus University of Thrace
Xanthi 67100 GREECE
e-mail bakalidi@xanthi.cc.duth.gr

A. TSOMPANOPoulos
Imaging Lab
Electrical Engineering Department
Democritus University of Thrace
Xanthi 67100 GREECE
e-mail tsompano@xanthi.cc.duth.gr

C. CHAMZAS* ASSOCIATE PROF.
Electrical Engineering Department
Democritus University of Thrace
Xanthi 67100 GREECE
e-mail chamzas@xanthi.cc.duth.gr

Abstract

JBIGW is a document archiving and retrieving system that is based on the ISO/IEC 11544 / CCITT T-82, encoding standard. This new lossless coding standard is much more suitable for image databases than the currently used CCITT group IV standard, since its additional capabilities include image database browsing, input output device independence and 50% better compression. The JBIGW bi-level image decoding, encoding and presentation program uses the Microsoft Windows version 3.1 graphical user interface software platform. The application supports the standard Windows MDI user interface in order to decode, present and process black and white images, that have previously been encoded using the JBIG encoding standard. The entire program has been developed using the Borland C++ ver. 3.1 compiler for the MS Windows system, and code generation for the 80386 microprocessor in order to achieve maximum decoding speed. The system in its current software implementation can decode images with speeds around 9600 bits/sec.

Introduction

JBIG is the new CCITT/ISO standard [1, 2] on "Coded Representation of Bi-level and Limited Bits-per-pixel Still Pictures" entitled "Progressive Bi-level Image Compression". This new standard has been developed by ISO/IEC JTC1/SC29/WG9 and it is suitable for lossless compression and progressive representation of images with a limited number of grey or colour values. Currently, the compression standards developed for Group 3 [3], and Group 4 facsimile are widely used beyond facsimile for the storage of images in the vast majority of image database and archival systems. These techniques were designed for compression of black/white text and line drawings, but, they perform poorly on images that have halftone or dithered content.
is a new technique, that is significantly more efficient than G3 and G4 on all types level images, while also providing new capabilities suitable for a mixed delivery document containing display terminals and printers. To do this, JBIG provides both a sequential mode for top down scan line compression and a progressive mode which can achieve higher compression and flexibility with the use of an adaptive technique for reducing the image data. In sequential mode, the JBIG coder uses characteristics of the image and automatically adjusts itself to efficiently encode it.

Initial applications of JBIG in its sequential form include its use as a new high compression scheme for traditional G3 and G4 facsimile. Other applications using higher compression and flexibility will use it as well. An example is the use of storage-retrieval and transmission of documents from a document data base. Application is the subject of this presentation.

The progressive capability that JBIG offers can make a significant improvement in the delivery of image databases and libraries [6] without impacting compression efficiency. JBIG provides the database with the capability, within a single compressed image, to display low resolution iconic version of the image for rapid browsing and to view the selected image at desired higher resolution for viewing on a display terminal or printing. JBIG can be used to store images in a suitable compressed format that allows the database to deliver images to a display and in a progressive manner from icon to the desired resolution. The image database delivers images to a display in a sequential manner building the image in mental images to the desired resolution, thus eliminating the need for a full page in the printer. The JBIG standard also documents a suggested algorithm for viewing high quality 2:1 reductions of the input image that is used for progressive compression.

Unlike for lossless encoding of colour and grey scale images on a bit plane basis both sequential and progressive delivery is also supported. The JBIG bit-plane encoding performs well over a wide range of bit levels. Certain applications, such as medical images, are particularly suitable for JBIG, because of their resolution and lossless requirements. Grey scale images or bi-level images with a mark-up overlays are other examples of images that would require this capability.

Under Windows (JBIGW)

In the following sessions we shall describe an implementation of JBIG under Windows from now on called JBIGW). This platform has been chosen due to its wide ability as a graphical user interface and due to the fact that it operates quite nicely on a large number of PC systems and I/O hardware configurations. Certain JBIGW implementation is based on a DOS application which was created by a previous JBIG implementation that run on a UNIX machine, to work on Intel's 80x86 family of microprocessors.

DOS-based code was used as the kernel around which Windows user interface elements were gradually developed. These elements include the Windows MDI user interface standard, common dialog boxes, standard BWCC controls and support for BICSI data interchange via the Window's Clipboard.

The following paragraphs contain a more detailed discussion of the aforementioned JBIGW features.

MDI (Multi Document Interface)

MDI is a standard user interface of the Windows operating system, which is used in cases where more than one "document" needs to be presented and processed within a single Windows application (See Figure 1).

An MDI application has a main window that is called the frame window in which the user can open and work with many documents, which in our case are the JBIG decoded images. The MDI application creates and controls a separate child window which contains each document. Looking at it from the user's point of view, MDI children windows are "normal" windows with the only difference that they cannot be moved

![Fig 1. JBIGW displaying the eight CCITT images at lowest resolution in tiled form outside the frame.](image)

An other advantage of the MDI user interface is that information can be very easily copied and pasted from one document to another, without having to create a second instance of the application. Copying and pasting inside JBIGW is done using the Windows's clipboard, which also allows the pasting and pasting of information to and from other Windows applications as well.

JBIGW fully supports all MDI functions including external scroll-bars for the frame window, title and cascade commands and a Window menu to help users select the image they want to work with, using the keyboard.

Common Dialog boxes for input and output are also supported to provide a known look and feel to experienced Windows users, while all user interface elements have been designed using Borland's BWCC (Borland Windows Custom Controls) widget library.

The Decoder

121
inside JBIG image decoding is performed using the JBIG W95-S1R1 (April 3 1992)
standard. JBIG's main feature is what is known as "progressive encoding, and
 decoding". Progressive decoding means that the user is given the ability to start with a
cry low resolution image and progressively decode it - doubling the image dimensions
each time - until a satisfactory resolution is finally reached. That way only the necessary
information is being decoded, thus reducing the overall decoding time. Enlarging an
image with the JBIGW program is done by just double-clicking with the mouse inside
the desired image, window or by selecting the "Enlarge" option from a pop-up menu that
appears inside each child window when pressing the right mouse button.

A small drawback of the current JBIG implementation is that the data for each image
level are stored in separate files, a fact that gives relative small decoding speeds
regarding the algorithm's capabilities.

The JBIG encoder is available under the application's File menu allowing the interactive
JBIG encoding of any existing bitmap that is located on the system's hard drives. The
program also allows the interactive encoding of the current image in cases when it has
been merged with an other during pasting.

The application allows conversions between the JBIG format and the MS-Windows 3.0
bitmap format, while support for other known image formats such as TIFF and PCX is
very soon expected.

Device Independence

Nowadays output devices offer a variety of resolutions ranging from a 75 dpi computer
screen to a 600 dpi PostScript laser printer. This means that various device resolutions
are available even inside the same computer system and even more, something - say a
Printed document - that looks satisfactory when displayed on the computer screen,
may look totally inadequate when printed, due to the fact that the printer's resolution
exceeds by far that of the screen. Most programs overcome this problem by scaling the
image so as to fit the output device's resolution. However this method does not provide
the best possible results, since the image becomes altered and disformed by the scaling
process.

The JBIG encoding method offers a different approach. Initially the image is scanned
and encoded using the highest possible resolution, but when it comes to display it, the
system can determine the output device resolution and further decode more information
or save user and system time by decoding only the necessary data for the display. That
way JBIG images do not need decoding up to maximum level in order to be displayed on
the screen, but the system can produce high resolution copies when a hard copy is
requested.

The Windows operating system, which is the platform for the JBIGW project offers all
the required information for each output device's capabilities, allowing a program to
decide how to react for each specific device. Eventually there may be an automatic
resolution specific decoding version but currently extra decoding is done manually.

Directory Browsing

Since the JBIG decoding for the first resolution level is done on the fly, at a speed of
about 8 Kbits/sec on an average 386 based machine, and 9 Kbits/sec on a 486, it is
possible to browse through system directories and bring up all the images contained in a

Fig 2 JBIGW's Interactive Encoder

Fig 3 The Browse dialog box showing current path and JBIG encoded images present

JBIGW offers a browsing tool (see Fig 3) that gives the application user a chance to have
a fast look at all the images present in his/her system and then interactively select and
decode the ones of any particular interest.

When the OK button is pressed on the dialog box of figure 3 all the images present in the
current path are immediately decoded and presented in tiled form as they are shown in
Figure 1.
On the other hand if nothing seems worth further decoding then the current directory may
be changed and the search for images be continued elsewhere.

JBIGW Decoding Speed

JBIGW was tested on a variety of images and computers. However, exact measurements
were taken by decoding a set of eight images, known as the CCITT group, which are the
images displayed in Figure 1. For this purpose three different computers were used,
using a relatively slow 8086 based machine running at 25 MHz with no cache
memory, to an 80386 running at 33 MHz with 128K of cache and finally to an 80486
machine running at 33 MHz with 256 KB of cache. For the evaluation purpose we used
80386 version of JBIGW, Results obtained on each computer for every image at every
color are shown in the chart of figure 4.

JBIGW is heavily dependent on disk speed, measurements can vary when taken on
cameras with the same clock speed but different hard drives. The measurements shown
were, were taken with the JBIGW tmp directory being mounted on a Novell 3.11 server
with a read speed of 400 KB/sec and write speed of 300 KB/sec.

The measurement shown above were taken using the version of JBIGW
which uses the 80386 instruction set. If the 80286 version is used then the above speeds
would be decreased by a factor of about 40%, on the same systems.

Finally we would like to mention that the current implementation of JBIGW, when run at
80486 based system, can decode images at a speed most of the times exceeding the
600 bits/sec speed rate currently used for most modems. This feature enables access to
remote databases in order to acquire and process JBIG encoded images.


D. Duttweller and C. Chamzas, "Probability estimation in arithmetic and

G. Story, L. O'Gorman, D. Fox, L. Schaper, and H. Jagdish, "The RightPages
Image Electronic Library for Alerting and Browsing", IEEE Computer, pp 17-25,


Conclusions and Future aspects

When the first version of JBIGW - then called ShowImage - was successfully
achieved decoding speed, and image presentation with a variable bit/pixel ratio, support
for Windows IDE and OLE features as well as support for local networks and document
base management.

References

1. CCITT Recommendation T.82 (1993), "Progressive Bilevel Image Compression".

   Compression".

3. CCITT Recommendation T.4, "Standardisation of Group 3 facsimile apparatus
document transmission".

4. J.H. Witten, R.M. Neal, and J.G. Cleary, "Arithmetic coding for data compresio