A Bilevel Document Archiving System, using JBIG\textsuperscript{3}

Authored by

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

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

C. CHAMZAS\textsuperscript{*} 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 Independency 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 9500 bits/sec.

\textsuperscript{3}This work was partially supported by EC under the program STRIDE-HELLAS #8

\textsuperscript{*} Address all correspondence to C. Chamzas
A MDI application has a main window that is called the **frame** window in which the user can open and work with many documents, in our case being the JBIG decoded images. The MDI application creates and controls a separate child window containing 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 outside the frame.

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 Window's clipboard, which also allows the pasting and porting of information to and from other Windows applications as well.

**Fig 1. JBIGW displaying the eight CCITT images at lowest resolution in tiled form**

JBIGW fully supports all MDI functions including external scroll-bars for the frame window, tile 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
files, a fact that gives relative small decoding speeds regarding the files.

User is available under the application's File menu or from the button on the system's hard drives. The program also allows the interactive encoding of the image in cases when it has been merged with another during pasting.

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

**Device Independency**

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 can be available even inside the same computer system and even more, something - say a bit mapped 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 gets scanned and encoded using the highest possible resolution, but when it comes to displaying 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 6 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 any one of them within a very small amount of time.

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 image present in the current path are immediately decoded and presented in tiled form as they are shown in
computers with the same clock speed but different hard drives. The measurements shown above, 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.

As mentioned earlier the measurements 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 a 80486 based system, can decode images at a speed most of the times exceeding the 9600 bits/sec speed rate currently used for most modem. This feature enables access to remote databases in order to acquire and process JBIG encoded images.

![Figure 4 Chart showing decoding speed of JBIGW in the CCITT image group on various computer systems](image)
Conclusions and Future aspects

When the first version of JBIGW - then called ShowImage - was successfully ran, approximately 2,000 lines of code were brought into service. At that time minimal operations were available. Currently the main system consists of more than 15,000 lines of code. The ongoing evolution of the application has been notably smooth, in the sense that new features were added, continuously enhancing, user interface elements or offering more Window's based features. Future improvements include further increase of the achieved decoding speed, and image presentation with a variable bit/pixel ratio, support for Windows DDE and OLE features as well as support for local networks and document data base management.

References

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


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


High-Speed Networking and Multimedia Computing

Arturo A. Rodriguez
Mon-Song Chen
Jacek Maitan
Chairs/Editors

8–10 February 1994
San Jose, California

Sponsored by

IS&T—The Society for Imaging Science and Technology
SPIE—The International Society for Optical Engineering

In cooperation with

IEEE Computer Society

Published by
SPIE—The International Society for Optical Engineering

Volume 2188

SPIE (The Society of Photo-Optical Instrumentation Engineers) is a nonprofit society dedicated to the advancement of optical and optoelectronic applied science and technology.