A testbed for the design of still-image codecs *

J. Oliver and M.P. Malumbres
Universidad Politécnica de Valencia. DISCA Department.
Camino de Vera 17, 46071 Valencia
E-mail: {joliver,mperez}@gap.upv.es

Abstract

Many image coder proposals and image compression standards have been developed recently [4], [6]. Hence, a performance evaluation tool is needed to determine the real benefits of these proposals. With this tool we should be able not only to design codecs that can combine the features of others previously published but also to design new codecs and evaluate their behavior before final implementation.

In this paper we present a general testbed for designing and testing whatever kind of image coder proposals. As a first approach, we have implemented an optimized version of the Shapiro's [7] EZW wavelet image codec. Also, as a reference, the JPEG standard library [2] was adapted to work in our testbed.

In order to verify the correctness of our codec implementation, and to show the behaviour of this tool, some simulation results were thrown in last section.

1 Introduction

Many image coder proposals and image compression standards have been developed recently [4], [6]. Although it would be useful to use only one general image compression standard, a growing number of standards were developed because of enhanced processing power, dedicated hardware and new compression techniques.

Sometimes it is difficult to choose the correct compression standard for a specific application. Some applications require fast real-time encoding, at the cost of the compression factor (video-conferencing), while other applications want maximum compression at encoding that do not need to be done in real-time, as long as decoding is really real-time (e.g. compressing a video stream on CD-ROM).

There are several software packages that are built for the evaluation of a specific kind of codec in order to carry out its standardization [2]. This kind of software is defined around a set of specifications previously defined, so it is difficult to reuse it for any other kind of image codec.

*Work partially funded by UPV

So, the main target of this paper is to build a performance evaluation tool that can easily accommodate whatever kind of image coder, so performance evaluation of different approaches could be done showing the benefits of each kind of codec in each possible application. We have included two codecs, a JPEG standard codec and our implementation of the EZW wavelet codec.

On the other hand, not only it is of interest to determine the real benefits of the existing proposals but to design new codecs and evaluate their behavior before final implementation as well.

This paper is organized as follows: In section 2 a description of the testbed for the design of image coders is given. Then, in section 3, a performance evaluation is performed comparing JPEG and an implementation of a wavelet EZW codec. Finally, in section 4 some conclusions and future work are drawn.

2 Performance Evaluation Tool Description.

In this section we define the main features of our tool and some details about its implementation.

The performance evaluation tool is composed of three main modules or units. The first one is the image format unit, which allows the definition of the image structure and the methods associated to it. It includes the definition of quality metrics and entropy expressions. Also there is a set of routines that import/export different standard image formats (BMP, PGM, PPM, YUV, etc.).

The I/O unit is responsible of the user interface. It defines two operation modes: interactive and batch. The former allows the user to introduce interactively the simulation parameters. The later mode is intended to be used without user intervention, giving the simulation parameters in an input file and supplying the results in a specific output file.

The most important unit is the image and video codec unit. It includes modules that implement the image and video codecs supported by the tool. Also there are generic modules that implement typical operations in most image and video codecs as arithmetic and Huffman coders, run-length coders, quantizers, etc. In order to allow the inclusion of codecs from other authors, a common interface was defined for both image and video codecs.

The evaluation tool presents the following features:

- Support for the development of image codecs. The tool supplies several modules that simplify the implementation of image codecs.

- Evaluation and simulation support. This tool allows the simulation and performance evaluation of different codecs. There are routines that compute some parameters typically used in the evaluation process like objective quality metrics (MSE, SNR, PSNR), first order entropy expression, compression rates, etc.

- It is modular, scalable and well documented. This allows other users to add new codecs, or part of one codec, and evaluate it quickly.
- System independent. All source code was written in ANSI C. This tool was intended for Windows and Linux platforms.
- Text user interface. As a consequence of the system independence, the user interface is text oriented. So, the input and output data are given by mean of text files. This allows us to maintain a database of evaluation results that can be accessed without rerun the simulations.

3 Preliminary Results

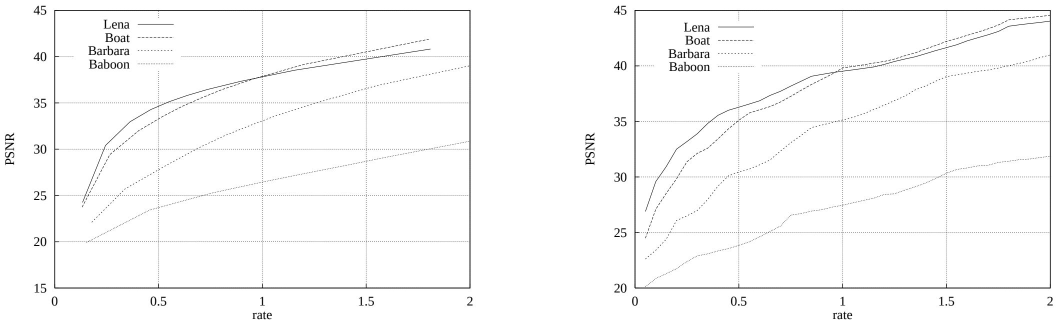
In this section, we present an evaluation of the JPEG standard implementation by the Independent JPEG Group [2] and our own version of the Shapiro's Zerotree codec [7]. We will use a set of standard images that are commonly used to evaluate image codecs. In particular, we have choosen for our evaluation experiments the Lena, Baboon, Barbara, and Boat images [1].

We have used the typical performance measures, the PSNR (measured in decibels) and the output bit rate (measured in bits per pixel) metrics. Also we will perform a simple subjective quality measure test comparing the same image encoded at the same bit rate with both codecs in order to show the picture quality differences visually.

When using JPEG we will use the JPEG quality factor with incremental steps of fixed size. The range of JPEG quality factor is from 0 to 100. We will use incremental steps of 10 to calculte the JPEG curves. In the case of EZW codec, we can obtain much more points in the rate/distortion curves as it is embedded.

If we compare curves from figure 1.a, we can see that Lena and Boat are the easiest images to be compressed by JPEG. Conversely, Babbon, full of high frequency details, is the image with the poorest rate/distortion relation. Figure 1.b shows a similar comparison using our Shapiro's EZW implementation instead of JPEG. Logically, results are comparatively similar

Figure 2 shows a comparison between JPEG and EZW when using the Lena and Baboon source images. It can be shown that the EZW algorithm outperforms JPEG along all the curve, increasing the differences for low target bit rates, in particular below 0.25 bpp.



(a) (b)

Figure 1: Rate/Distortion curves for (a) JPEG and (b) EZW.

Also, we have evaluate the importance of choosing a good filter to implement the wavelet transform in our EZW implementation. Several filter banks were selected to perform this experminet. In particular, we have choosen the well-known Daubechies 4-tap filter (fast and easy implementation), Biorthogonal 9/7 filter (used in most wavelet codecs), Villasenor 10/18 filter and the filter used by the original EZW. Several simulations were run using the Baboon image, each of the above mentioned filters and the EZW implementation. As shown in Figure 3 the behaviour of the filters is quite similar with the exception of Daubechies 4-tap filter that shows the worst results and the Villasenor filter that seems to be the best results. This is due to the filter ability to compact energy in the low frequency bands.

Our performance evaluation tool also allows subjective quality tests by supplying the decoded version of the original image using the same bit rate or the same PSNR from both codecs. In Figure 4 we show the decoded versions of the original Baboon image of JPEG and EZW at 0.2 bpp (i.e. both compressed versions have the same size). As it can be seen, the JPEG decoded version shows blocking artifacts that are due to its DCT transform stage. These artefacts significantly degrades the perceived image quality when compared with the EZW decoded version.

4 Conclusions and future work

In this paper, we have developed a general testbed for the design of still-image codecs, this tool is able to evaluate whatever kind of image codec. It was designed taking in mind its modularity, scalability, and portability, defining common interfaces that allow other users to add new codecs or pieces of codecs in order to know its behavior. The tool includes a lot of test images that are commonly used in image research

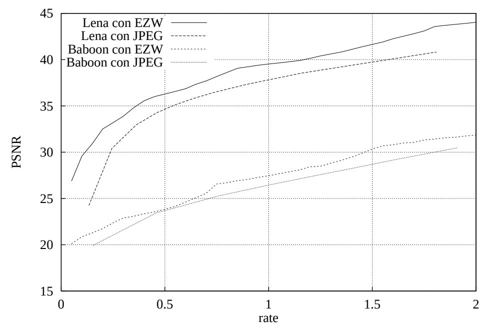


Figure 2: JPEG and EZW comparison (Lena and Baboon images).

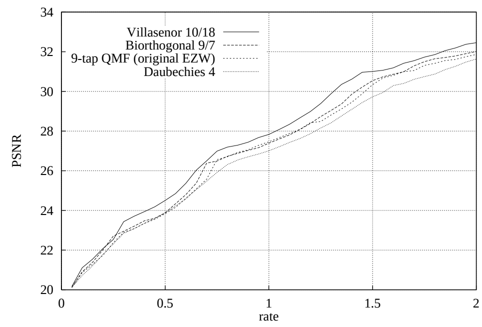


Figure 3: Evaluating different filters (Baboon image).

works, and also two image codecs: JPEG and our implementation of an EZW wavelet image codec.

In order to check the correctness of our tool and the supported codecs, several test were run. Among them, a comparison between both codecs was performed using typical performance mesaurements and a subjective quality test. The subjective quality test shows the differences in perceived quality of both codecs, showing the poorest behaviour of JPEG, specially at low bit rates. Finally, we have showed the importance of choosing a good filter bank to achieve the best performance results.

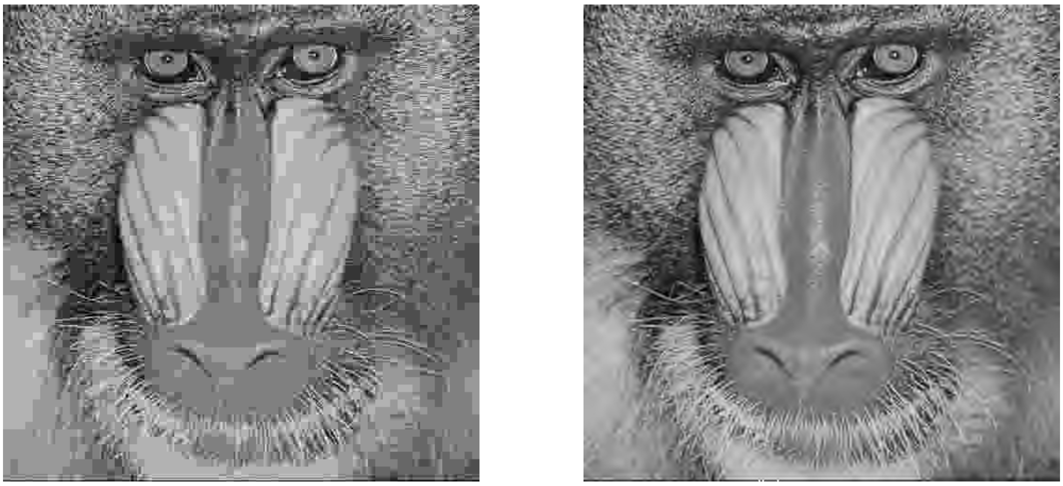
As future work some improvements of our tool are planned. Among them, we plan to include more standards and video codecs like MPEG [5] and H.263, and new versions of wavelet-based codecs.

References

[1] CityU-UPL. Image Database. http://www.image.cityu.edu.hk/imagedb

[2] Independent JPEG Group. JPEG software version 6b. ftp://ftp.uu.net/graphics/jpeg/, 1998.

[3] Joint Photographic Expert Group. Digital Compression and Coding of Continuous Tone Images (Part 1: Requirements and Guidelines). ISO/IEC 10918-1, 1992



(a) (b)

Figure 4: Subjective quality test of Baboon image coded with (a) JPEG and (b) EZW at 0.2 bpp.

[4] D. Lee. JPEG 2000: Developments in still-image coding system. Presentation at EUSIPCO98 (http://eurostill.epfl.ch/eusipco-j2k-pres/sld001.htm).

[5] Motion Picture Experts Group. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s. ISO/IEC 11172-2, 1993

[6] RICOH Silicon Valley, Inc. RICOH CREW Image Compression Standard. Version 1.0 (Beta), April 1999

[7] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. on Signal Processing, vol41, n12, December 1993