INTELLIGENT COMPRESSION OF STILL IMAGES   C115

Frédéric GILBERT
ALCATEL ALSTHOM RECHERCHE
Route de Nozay
91460 MARCOUSSIS
FRANCE
Frederic.Gilbert@aar.alcatel-alsthom.fr

Fabien MOUTARDE
ECOLE DES MINES DE PARIS
60, Bd Saint-Michel
75272 PARIS Cedex 06
FRANCE
moutarde@cc.ensmp.fr

Abstract

We present here a framework and prototype for "intelligent" compression of still images, which takes advantage of a set of algorithms (such as JPEG, wavelets, fractals, ...) by switching between them according to the image type and content, and target compression ratio; our intelligent compressor also tunes automatically the possible algorithm-specific parameters (e.g. JPEG quantization factor) in order to meet the user compression objective. Our coder proves to be successful with probability ≈ 70% in selecting automatically the best technique, and the difference between the target compression ratio and the obtained one is lower than 10%.

1. Motivations

As more and more digital images are used for various purposes, their compression has become an important issue, and an increasing number of persons have to be able to use and tune the different algorithms. To allow non expert people to do this, it is useful to have an "intelligent" compression scheme choose the algorithms and its parameters from the image characteristics and the user objective.

Another motivation for our intelligent compression approach is the upcoming end of the monopoly of the JPEG [1] standard. New compression techniques, based on wavelets [2] and fractals [3], can now outperform JPEG, particularly at high compression ratios. Graphical images are also often better compressed using GIF. No compression algorithm proving to be the ultimate one, it is useful to adapt the compression process to the image content. This method varies from other existing adaptive schemes which modify some transform or quantization characteristics (as DCT quantization tables for JPEG [4]) according to the image content.

2. Intelligent coder design

The intelligent coder was built from a set of algorithms that we thought complementary from an application range point of view: DCT based JPEG [1], a wavelet based algorithm with zerotree coding [2], a fractal based algorithm [3] for lossy compression; and GIF and Lossless JPEG [1] for lossless compression.

Our smart coder is composed of these 5 compression routines and of an adaptive module, which can be decomposed into a color counting module, an image complexity criteria computation module, and a selector function. The selector chooses the algorithm to use and its parameters from the image characteristics and the user objective (fig. 1).

The image complexity is evaluated through the criteria given in formula (1). The number of colours is used, in the case of lossless compression, to distinguish continuous tone images (preferably compressed with JPEG) and graphical images (preferably compressed with GIF). The reconstruction error measure used to evaluate compression performance is the Quadratic Mean Error (QME).

The selector itself is obtained through an experimental calibration process, which consists in testing all algorithms in competition on a database of images and for various compression ratios, in order to build a performance model for each algorithm. These tests allow to create a performance model of algorithms, in the form of abacuses.

3. Intelligent coder calibration

The 5 algorithms were applied to a calibration database of 50 images, the lossless ones being applied once for each image, and the lossy ones with evenly spaced values of their parameter.

Compressed image complexity, parameter value, obtained compression ratio, and (for lossy algorithms) QME were stored in the form of 2D (fig. 2) or 3D (fig. 3) abacuses, then smoothed applying smoothing splines to experimental points.

For algorithms (like JPEG) that are not directly tunable with a compression ratio or a QME, other abacuses are created to model, for each algorithm, the QME or compression rate as a function of image complexity and parameter value (fig. 4).

Once calibration have been done, the selection acts as follows:

- Algorithm choice: look up abacuses to find, for an image of a given complexity, which algorithm will provide the lowest QME (resp. the highest compression ratio) for a required target compression ratio (resp. QME).
- Parameter determination: look up chosen algorithm abacus to find the parameter which, given the image complexity, is expected to respect the compression objective.

4. Results

The adaptive coder was tested on another set of 50 images. Results show that 1/ when an algorithm is chosen by the coder to compress an image, it is actually the best one in 70% of cases; 2/ when an algorithm is the best among others, it is actually chosen by the coder in 68% of cases. It can particularly be seen that the fractal based algorithm is never chosen, and that the coder tends to under-estimate JPEG vs. wavelets, which causes JPEG to be chosen less often than it should be. A secondary result is that, in the opposite of what is found in [5], wavelets do not prove to be always better than JPEG; among others, this can be due to differences in Shapiro algorithm implementation.

Another important result is the good precision with which it fulfills the compression objectives (fig. 5 and 6): the maximum error between target and obtained values are about 10% for compression ratio with JPEG, 20% for QME with JPEG, and 12% for compression ratio with Lossless JPEG.

5. Discussion

The mean performance of the algorithm selection is rather good (≈70%), which allows to validate he method, but the error rate is still too high when JPEG is selected or should be selected. Some improvements can be envisaged to improve performance, the main ones being changing the smoothing method, changing the reconstruction error and the image complexity measure, using a more specialized calibration database, and adding more algorithms to the coder.

Particularly, as the QME is not really taking in account the visual aspect of images, tests were made using a measure integrating a psycho-visual criteria [6]. The computation cost and numerical instability prevented us to keep this solution.

If our adaptive compression scheme cannot be used for color images yet, it could be easily modified to apply the same treatments on each components of color images. However, it could be interesting to define some global complexity measure for color images, so as to process them globally.

6. Conclusion

Our adaptive compression scheme allows to optimally compress an image (in the sense that it selects the best algorithm to attain a compression objective), in an user-friendly way (in the sense that algorithm choice and parameterization is transparent for the user), for a computation cost overhead nearly negligible.

It proves to have interesting performances that make the approach promising and susceptible to be improved in some simple ways.

Bibliography

[1] ISO-IEC/JTC1/SC29/WG1, «*Digital compression and coding of continuous-tone still images: requirements and guidelines*», ISO/IEC 10918-1, February 1994.

[2] J. Shapiro «*Embedded Image Coding Using Zerotrees of Wavelet Coefficients*», IEEE Trans. on Signal Processing, Vol. 41, n°12, Dec. 1993.

[3] Y. Fisher (ed.), «*Fractal image compression: theory and applications*», Springer Verlag, New York, 1995.

[4] A.B. Watson, «*Visually optimal DCT quantization matrices for individual images*«, proceedings of Data Compression Conference '93, held in Snowbird (USA), March 1993.

[5] V.R. Algazi & R.R. Estes Jr, «*Comparative performance of wavelet and JPEG coders at high quality*», proceedings of SPIE conference on Very High Resolution and Quality Imaging, held in San Jose (USA), February 1997.

[6] N.B. Nill & B.H. Houzas, «*Objective Image quality Measures derived from digital image power spectra*», Optical Engineering, Vol. 31, No 4, pp 813-825, April 1992.

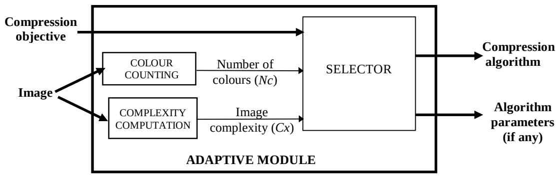

Figure. 1

$$C_x = \frac{1}{(w-1)\times h}\sum_{i=1}^{h}\sum_{j=1}^{w-1}\left|x_{i,j} - x_{i,j+1}\right|$$

formula (1) : image complexity estimation, where x_{ij} = pixel value at line i and column j, and (w,h) = image dimensions

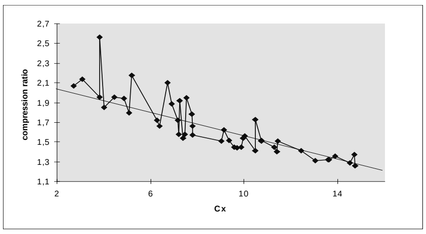

Figure 2: compression_ratio = f(image_complexity) for Lossless JPEG

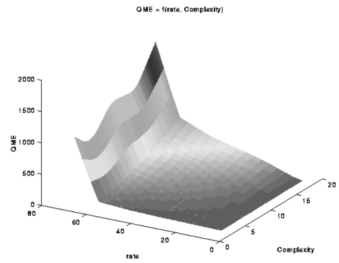

Figure 3: QME = f(compression_ratio, image_complexity) for JPEG

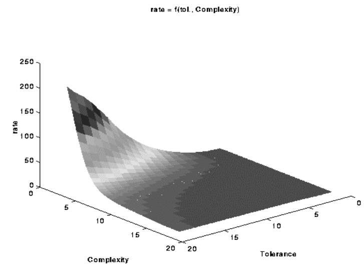

Figure 4: compression_ratio = f(tolerance_parameter, image_complexity) for Fisher's IFS

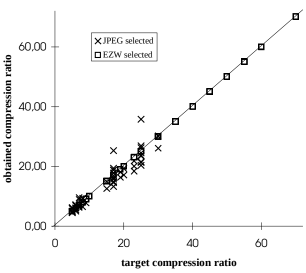

Figure 5: comparison of target and obtained compression ratios for various test images and ratios

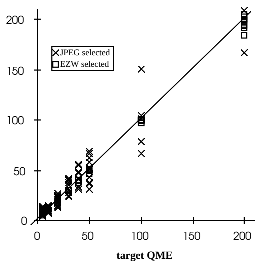

Figure 6: comparison of target and obtained QME for various test images and QMEs.