Evolved Nonlinear Predictor Functions for Lossless Image Compression

Kevin M. Barresi
Stevens Institute of Technology
1 Castle Point Terrace
Hoboken, NJ, United States of America
kbarresi@stevens.edu

ABSTRACT

Due to the increased quantity of digital data, especially in the form of digital images, the need for effective image compression techniques is greater than ever. The JPEG lossless mode relies on predictive coding, in which accurate predictive models are critical. This study presents an efficient method of generating predictor models for input images via genetic programming. It is shown to always produce error images with entropy equal to or lower than those produced by the JPEG lossless mode. This method is demonstrated to have practical use as a real-time asymetric image compression algorithm due to its ability to quickly and reliably derive prediction models.

Categories and Subject Descriptors

I.4.2 [**Image Processing and Computer Vision**]: Compression (Coding); I.2.m.c [**Artificial Intelligence**]: Methodologies—*Evolutionary computing and genetic algorithms*

Keywords

Genetic Programming; Image Compression

1. INTRODUCTION

Lossy image compression results in irreversible loss of image data and can introduce visual artifacts. To remedy this, lossless compression methods can be used. While a large number of lossless techniques exist [1–4], there is significant room for improvement in terms of compression performance. Attempts at providing lossless compression via evolutionary algorithms have proven unrealistic for real-time use and do not provide strong compression ratio improvements [5, 6].

Predictive coding is a simple, yet effective lossless method for reducing the entropy of an image. With lower entropy levels, the encoded data is more easily compressed with an entropy coder. Its main component, the predictive model, uses the value of an individual pixel as compared to the values of its neighbors. The differences between the predicted pixel values and actual pixel values are stored in an *error image*. If the prediction function is perfectly accurate, the resulting error image will contain all zeroes, and only the input image's border pixels will need to be stored in order to perfectly reconstruct the image. On the other hand, a poor model will result in irregular patterns of high and low pixel values resulting in high entropy, and thus, a lowered ability to be compressed via an entropy encoder.

The JPEG lossless mode utilizes seven static prediction functions [4]. During compression, the model that provides the most accurate prediction is selected and used to encode the image. Naturally, seven prediction models alone cannot provide optimal predictions for the infinite range of possible input images.

We propose a method of evolving nonlinear prediction models in a practical way. The proposed method is demonstrated to result in error images with an average 3.84% lower entropy than the JPEG lossless mode, while at the same time maintaining resource consumption and runtimes consistent with that of conventional image compression techniques. Empirical results show that the proposed method is able to compress on the order of two seconds per image.

2. PROPOSED METHOD

The proposed method uses a population of candidate prediction models. Each individual is initially seeded with one of the seven JPEG lossless mode prediction functions. An elitist selection scheme is used to guard against population regression. Throughout generations, the population is maintainted at a constant size. Three types of mutations are used: point, subtree insertion, and subtree deletion. Mutations are introduced randomly during each generation. The optimal mutation probability and population size was determined empirically to be 15% and 100, respectively.

The prediction models are represented as s-expressions, where chains of operators and variables are fashioned as a tree structure. The expression tree consists of a number of *functions* and *terminals*. Each function describes an operation, and each terminal acts as a tree leaf, representing some constant number. The proposed method uses the following functions and terminals: addition, subtraction, multiplication, division, minimum/maximum, left pixel, top-left pixel, top pixel, top-right pixel, and assorted constants (1, 2, 5, 10, 100).

Prediction model fitness is judged by the resulting error image's entropy of probability distribution. As shown in Equation 1, the fitness value is scaled based on the worst-case entropy level of the given image. The resulting fitness

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-1964-5/14/07.
http://dx.doi.org/10.1145/2598394.2598503 .

Table 1: Mutation probability versus percent entropy reduction

Mutation probability	Average entropy improvement
5%	2.43%
10%	3.67%
15%	**3.84%**
20%	3.58%
25%	3.53%
30%	3.04%

Table 2: Generations versus percent entropy reduction

Generations	Average entropy improvement
50	3.84%
100	4.86%
200	5.08%
300	5.13%
500	5.21%

for the i^{th} individual $F(i)$ is higher for better performing models, and lower for poorer models.

$$F(i) = 1 - \frac{-\sum_j p_j log_2 p_j}{log_2 N} = 1 - \frac{Entropy}{WorstCaseEntropy}. \quad (1)$$

$$P_b(i) = \frac{i-1}{N-1}. \quad (2)$$

Using the raw fitness value, all individuals are ranked using a common linear ranking system. This scaling method is demonstrated to perform better than other ranking systems in terms of resilience to disruption [7]. Using normalized fitness values, the i^{th} individual's probability of breeding $P_b(i)$ is calculated using the formula shown in Equation 2.

A major limiting factor in evolutionary based image compression is speed [5], so the proposed method introduces several performance enhancements. Parallelization was used extensively during fitness evaluation. In practice, runtimes were decreased by a factor of three using a four core computer with multithreading. Error entropy is directly related to compressed image size. Exploiting this relationship, fitness is based on error image entropy rather than compressed image size. This shortcut results in significant runtime reduction without affecting prediction model quality.

3. EXPERIMENTAL RESULTS

The proposed method was implemented as a C++ application on a 3.3 GHz, four core processor. Grayscale images were tested; it is a trivial extrension to apply this method to color images.

Table 1 shows the average entropy reduction over the best JPEG predictor. Various mutation probabilities were tested on an ImageNet dataset consisting of 500 images. Populations ran for 50 generations each, with each population consisting of 100 individuals. Runtimes averaged at 2.5 seconds per image. It was found that a 15% mutation probability provided the best entropy improvement, consistent with the high mutation probabilities often used in image processing applications.

In comparison, Table 2 shows average entropy improvements for an increasing number of generations. All populations used the optimal 15% mutation probability shown in Table 1. These results suggest that longer runtimes do improve evolved prediction models However, populations lasting longer than 50 generations resulted in runtimes too long to be considered practical. For example, a 100 generation population resulted in runtimes averaging slighly more than one minute per image.

From these results, it was concluded that the best population parameters for real-time compression are 50 generations, 100 individuals, and 15% mutation probability. These parameters result in an average 3.84% entropy reduction over the best JPEG prediction model, while maintaining average compression runtimes of 2.5 seconds.

4. CONCLUSIONS

Results from this study show that GP can be applied to image compression in a real-time manner that competes with conventional compression methods in terms of runtime and compression ratio. Compression improvements are an average 3.84% better than the JPEG lossless mode. The proposed method also has the benefit of being an asymmetric compression algorithm.

While the results from this study are certainly encouraging, there are still several points of interest that can be improved. In particular, these findings suggest that the next step is to apply similar evolutionary techniques to higher order prediction models utilizing non-local pixels. Recent work also suggests that deriving prediction models for image subregions holds promise for higher compression ratios.

5. REFERENCES

[1] M.J. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *Image Processing, IEEE Transactions on*, 9(8):1309–1324, 2000.

[2] Xiaolin Wu and N. Memon. CALIC - a context based adaptive lossless image codec. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 4, pages 1890–1893 vol. 4, 1996.

[3] P.W. Katz. String searcher, and compressor using same, September 24 1991. US Patent 5,051,745.

[4] G.K. Wallace. The JPEG still picture compression standard. *Consumer Electronics, IEEE Transactions on*, 38(1):xviii–xxxiv, 1992.

[5] Alex Fukunaga and Andre Stechert. Evolving nonlinear predictive models for lossless image compression with genetic programming. In *Programming, Proc. of the 3rd Annual Genetic Programming Conf*, pages 95–102, 1998.

[6] A. Kattan and R. Poli. Evolutionary synthesis of lossless compression algorithms with GP-ZIP3. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, 2010.

[7] Thomas Bäck and Frank Hoffmeister. Extended selection mechanisms in genetic algorithms. pages 92–99. Morgan Kaufmann, 1991.