Effective Image Compression using Evolved Wavelets

Uli Grasemann
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712
uli@cs.utexas.edu

Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712
risto@cs.utexas.edu

ABSTRACT

Wavelet-based image coders like the JPEG2000 standard are the state of the art in image compression. Unlike traditional image coders, however, their performance depends to a large degree on the choice of a good wavelet. Most wavelet-based image coders use standard wavelets that are known to perform well on photographic images. However, these wavelets do not perform as well on other common image classes, like scanned documents or fingerprints. In this paper, a method based on the coevolutionary genetic algorithm introduced in [11] is used to evolve specialized wavelets for fingerprint images. These wavelets are compared to the hand-designed wavelet currently used by the FBI to compress fingerprints. The results show that the evolved wavelets consistently outperform the hand-designed wavelet. Using evolution to adapt wavelets to classes of images can therefore significantly increase the quality of compressed images.

Categories and Subject Descriptors

G.1.2 [**Numerical Analysis**]: Approximation—*Wavelets and Fractals*; I.4.2 [**Computing Methodologies**]: Image Processing and Computer Vision—*Compression (Coding)*; I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*; G.1.6 [**Numerical Analysis**]: Global Optimization

General Terms

Algorithms, Experimentation, Performance

Keywords

Wavelets, Image Compression, Lifting, Genetic Algorithms, Coevolution

1. INTRODUCTION

Image compression is one of the most important and successful applications of the wavelet transform. Mature wavelet-based image coders like the JPEG2000 standard [15] are available, gaining in popularity, and easily outperform traditional coders based on the discrete cosine transform (DCT) like JPEG [25].

Unlike in DCT-based image compression, however, the performance of a wavelet-based image coder depends to a large degree on the choice of the wavelet. This problem is usually handled by using standard wavelets that are not specially adapted to a given image, but that are known to perform well on photographic images.

However, many common classes of images do not have the same statistical properties as photographic images, such as fingerprints, medical images, scanned documents, and satellite images. The standard wavelets used in image coders often do not match such images, resulting in decreased compression or image quality. Moreover, non-photographic images are often stored in large databases of similar images, making it worthwile to find a specially adapted wavelet for them. As Chris Brislawn, one of the architects of WSQ [13], the FBI's standard for fingerprint compression, states [2]:

> "Choosing wavelets for image coding applications is still a somewhat inexact science, depending on a lot of trial and error. There are a few "standard" wavelet families [...] that seem to work well for image coding, although that is not a task for which they were specifically designed. In the future we hope to be able to design wavelets (or wavelet-like filter banks) that are optimized for a specific application, like fingerprints. Until then we'll probably stick with proven performers [...]."

In this paper, a coevolutionary genetic algorithm based on *Enforced Sub-Populations* [11, 9] and a mathematical technique called *Lifting* is used to find wavelets that are specially adapted to a particular class of images. The approach is tested in the fingerprint compression domain, which provides a systematic comparison to other current approaches. The wavelets found by the GA are tested in a state-of-the-art image coder, and compared with standard wavelets, including the winner of a competition held by the FBI to find the best wavelet for fingerprint compression [13]. The evolved wavelets turn out consistently better, demonstrating that evolutionary discovery can outperform human design in an important task.

The remainder of this paper is structured as follows. The

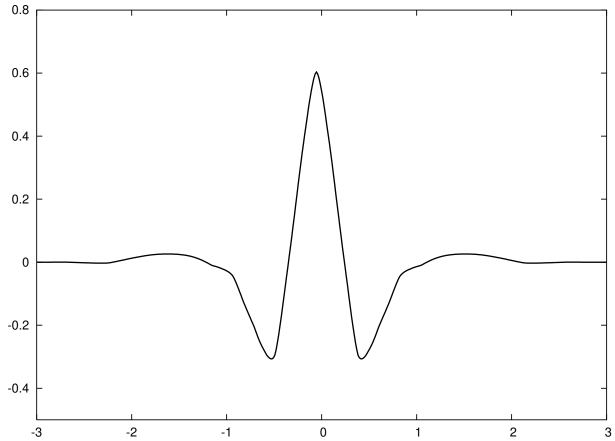


Figure 1: An example wavelet function. This wavelet is the Antonini 9/7 wavelet [1], also known as the FBI wavelet, because the FBI uses it to compress fingerprints [13].

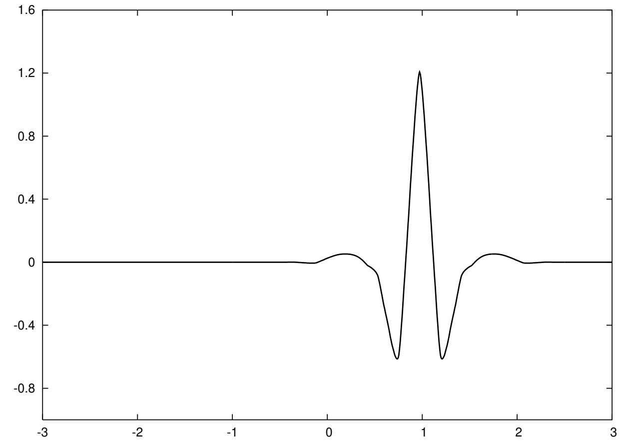


Figure 2: A translated and dilated version of the Antonini wavelet in figure 1. The Discrete Wavelet Transform (DWT) expresses the input data as a weighted sum of such translates and dilates.

next section gives a brief tour of wavelet-based image compression. Section 4 describes the algorithm and the evaluation function used. Section 5 reports experimental results, and section 6 discusses future directions for the wavelet evolution approach.

2. BACKGROUND

Both wavelet theory and wavelet-based image compression are complex and evolving subjects. This section gives a brief high-level overview of these topics. For more details on classical wavelet theory, see [14]. In addition, [6] contains an introduction to lifting, and [7] covers the basics of wavelet-based image compression.

2.1 Wavelets

Wavelets are a mathematical tool for representing and approximating functions hierarchically. At the heart of wavelet theory, there is a single function ψ, called the *mother wavelet*. Any function can be represented by superimposing translated and dilated versions of ψ, denoted by $\psi_{j,i}$, where i and j are the translation and dilation parameter. We are focusing on the discrete case where i and j only take on integer values. The $\psi_{j,i}$ can be computed from the mother wavelet as

$$\psi_{j,i}(x) = 2^{\frac{j}{2}}\ \psi(2^j x - i). \tag{1}$$

Figure 1 shows an example wavelet, and figure 2 shows a translated and dilated version of that wavelet.

All the translates of ψ for a specific dilation j span a function space W_j:

$$W_j = span\{\ \psi_{j,i} \mid i \in \mathbb{Z}\}. \tag{2}$$

The W_j are called *wavelet spaces* or *detail spaces*, because each of them adds a level of detail to the wavelet representation of a function. All of the detail spaces combined form a basis in which any function can be expressed.

The process of decomposing a function into *wavelet coefficients* (a scaling factor for each of the $\psi_{j,i}$) is called *wavelet transform*. If the parameters i and j take on dicrete values, we have a *discrete wavelet transform* or *DWT*, essentially leading to a finite number of coefficients.

In order to compute the DWT of a function f, we need to find one wavelet coefficient $\gamma_{j,i}$ for each $\psi_{j,i}$, such that

$$f = \sum_{j,i} \gamma_{j,i}\psi_{j,i}. \tag{3}$$

If a wavelet basis (i.e. the set of all $\psi_{j,i}$) is *orthogonal*, then the $\gamma_{j,i}$ are given by

$$\gamma_{j,i} \ = \ \langle f, \psi_{j,i}\rangle \ = \ \int_{-\infty}^{\infty} f(x)\overline{\psi_{j,i}}(x)dx, \tag{4}$$

where the bar denotes the complex conjugate. Otherwise, a *dual* wavelet $\widetilde{\psi}$ is necessary such that ψ and $\widetilde{\psi}$ together are *biorthogonal*, which basically means that the transform must be invertible. We can then use $\widetilde{\psi}$ for determining the wavelet coefficients (equation 4), and the original wavelet for the inverse DWT (equation 3). Note that an orthogonal wavelet is just a special case of a biorthogonal one where $\widetilde{\psi} = \psi$.

2.2 Filters and the Fast Wavelet Transform

Computing a wavelet transform in the way just described is expensive and cumbersome. However, an algorithm called the *Fast Wavelet Transform* or *FWT* allows computing the wavelet coefficients by recursively applying a pair of digital filters to the data, much like the Fast Fourier Transform reduces a discrete fourier transform to computing a few finite sums.

A *digital filter* can be defined by giving a sequence of real numbers called *filter coefficients*. It is applied by convolution with an input sequence. A filter is said to have *finite impulse response (FIR)*, if its coefficients are non-zero only on a finite range. A FIR filter can be represented by a finite number of coefficients and the index of the leftmost non-zero coefficient.

The filter pair used in the FWT uniquely determines the mother wavelet ψ and also (in the biorthogonal case) the dual wavelet $\widetilde{\psi}$. In order to define a valid wavelet transform, a filter pair must be *complementary*, which is the same as saying that the associated wavelet must be biorthogonal.

Ensuring that a filter pair is complementary and that the individual filters are finite are the basic contraints in wavelet design.

2.3 Lifting

The Lifting scheme, introduced by Sweldens [22] in 1996, offers an effective way to construct complementary filter pairs. A finite filter, called a *lifting step*, is used to generate a new filter pair from an existing pair. Multiple lifting steps can be applied consecutively. In [6], Sweldens and Daubechies proved two important properties of lifting:

- Lifting preserves biorthogonality, i.e. if the original filter pair is complementary, then so is the new pair, no matter what lifting step is applied.
- Any wavelet with finite filters can be expressed as a sequence of lifting steps. Starting with the trivial wavelet transform (called the *Lazy Wavelet*), all possible wavelets can be reached by applying a finite number of finite-length lifting steps.

These two properties make lifting a powerful tool for constructing new wavelets: Starting from a known complementary filter pair, other complementary pairs that are better adapted to the task at hand can be generated by applying lifting steps.

2.4 Wavelet-based Image Compression

All modern image coders are *transform coders*, i.e. they have the structure shown in figure 3.

Transform coders first apply an invertible transform to the image data in order to decorrelate it. Examples of such transforms are the discrete cosine transform (the basis for JPEG compression), and the discrete wavelet transform, the basis for JPEG2000 and other wavelet coders. The performance of a transform coder depends largely on how well the transform decorrelates the signal. A well decorrelated signal consists mainly of coefficients close to zero.

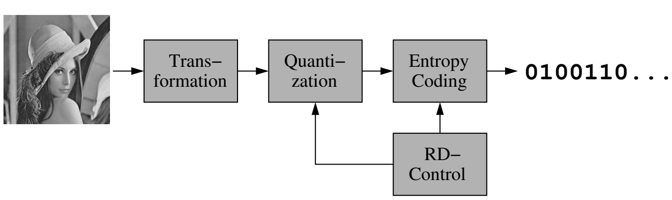


Figure 3: The structure of a transform coder. The signal is first decorrelated using an invertible transform, then quantized and entropy coded. The rate-distortion (RD) unit controls the quantization to minimize the distortion within the available bit rate. The performance of a transform coder depends on how well the transform decorrelates the image data.

After the transform step, the coefficients are quantized, i.e. expressed using symbols from a finite alphabet, and entropy coded, using as little space or bandwidth as possible. The rate-distortion (RD) unit controls the quantization in order to achieve minimal distortion within the available bit rate.

All three steps of an image coder have an impact on the image quality achieved for a given compression ratio. In particular, a better transform results in better compression performance.

3. RELATED WORK

Adaptive wavelet bases have been an active research area since the early 1990s. Traditional approaches are based on *dictionary methods*, where a basis is selected from an overcomplete set of predefined functions called *atoms*. Examples of such methods are the *best basis* algorithm [5] and *wavelet packets* [26]. In [16] and [18], evolutionary algorithms are used for adaptive dictionary methods. Dictionary methods are in a sense orthogonal to the approach used in this paper, and could be easily combined with it.

Several stochastic optimization techniques have been applied to the design of wavelets. Monro and Sherlock [20] use simulated annealing to find wavelets with balanced uncertainty in space and frequency. Hill et al. [12] used a genetic algorithm to optimize the parameters of a windowed trigonometric function that can be used in a continuous wavelet transform.

The lifting technique has provided new ways to adapt wavelets to the data being transformed. The common approach is to choose locally between a class of lifting steps while transforming an image. For example Claypoole et al. [3] used this approach to adapt wavelets to a given signal by optimizing data-based prediction error criteria.

Genetic algorithms, combined with the lifting technique, offer a way to find a single wavelet specifically adapted to a class of images based on real-world compression performance. A first version of such an algorithm was introduced in [11], and applied to a simple signal compression problem. In this paper, the method is shown to be effective in the more challenging task of compressing fingerprint images.

4. EVOLVING WAVELETS

In section 2.3, two useful properties of lifting were mentioned:

- Lifting preserves biorthogonality, and
- any wavelet can be expressed as a sequence of lifting steps.

These two properties make sequences of lifting steps an effective representation for wavelets in a genetic algorithm, because (1) any random sequence of lifting steps will encode a valid (i.e. biorthogonal) wavelet, and (2) any wavelet can be represented using the genetic code. In this section, a coevolutionary genetic algorithm that evolves wavelets encoded as lifting steps will be described.

Algorithm

The coevolutionary GA used is closely related to the *Enforced Sub-Populations (ESP)* neuroevolution algorithm introduced by Gomez and Miikkulainen [9]. ESP evolves a number of populations of individual neurons in parallel. In the evaluation phase, ESP repeatedly selects one neuron from each sub-population to form candidate networks. The fitness of a particular neuron is the average fitness of all networks in which it participated.

This concept can be easily applied to wavelet evolution: Several populations of lifting steps are evolved in parallel, Wavelet-ESP

input: N, the number of sub-populations
L, the lengths of the lifting filters
M, the size of each sub-population
P, the mutation rate

1. Initialize
Create M filters of length L for each of the N sub-populations, and randomize them.

2. Evaluate
Select N lifting steps, one from each sub-population, and evaluate the resulting wavelet. Add the fitness to the cumulative fitness of all participating steps. Repeat until each step has been evaluated 10 times on average.

3. Recombine
Rank the lifting steps in each sub-population by their average fitness. Each step in the top quartile is recombined with a higher-ranking step. The offspring is mutated with probability P and replaces the lowest-ranking half of each sub-population.

4. Repeat
Repeat the Evaluate-Recombine cycle for a fixed number of generations.

Figure 4: The ESP algorithm applied to wavelets. Several populations of lifting steps are evolved in parallel, and are combined in the evaluation phase to form wavelets.

and are randomly combined to form wavelets, which are then evaluated. No migration or crossover occurs between sub-populations. Figure 4 describes the algorithm in detail.

Representation

A lifting step is represented as a fixed-length sequence of floating point numbers for the filter coefficients, and a single integer for the leftmost index of the filter. Using a fixed number of fixed-length steps limits the number of wavelets that can be represented. However, it also limits the length of the wavelet filters, which is a desirable effect. Also, most wavelets used in practice can be factored into a small number of short lifting steps [6], so this limitation is unlikely to interfere with finding good solutions.

Initialization

Each chromosome is initialized by setting the values of the coefficients to random values from a gaussian distribution with mean 0 and variance 0.5, and setting the leftmost index of each filter to a random integer between -2 and 2. These settings reflect the values commonly found in lifting steps.

Crossover

The crossover operator performs simple one-point crossover on the coefficients. The integers representing the leftmost indices of the parent filters are randomly assigned to the children.

Mutation

A chromosome is mutated by adding low-variance gaussian noise to a random filter coefficient and/or adding ± 1 to the integer representing the leftmost index.

Fitness Evaluation

In image compression, the ideal measure of fitness would be the performance in an actual transform coder as described in section 2.4. However, there are two problems with this approach. First, evaluating a wavelet using a transform coder is prohibitively expensive. Second, in order to make a fair comparison between two wavelets, either the available number of bits needs to be fixed and the resulting distortion used as a fitness measure, or vice versa. Both options are inexact and expensive for actual transform coders.

Evaluation Function

input: D, the input data
W, a candidate wavelet
R, the compression ratio

return: The fitness of W.

1. Transform
Transform D using the wavelet W.

2. Compress
Sort the resulting wavelet coefficients. Keep only the largest $R \times |D|$. Set the rest to zero.

3. Reconstruct
Perform an inverse transform using W and the altered wavelet coefficients.

4. Measure the Image Quality
Measure the resulting image quality (peak signal to noise ratio), and return it.

Figure 5: The evaluation function is an idealized version of a transform coder: Instead of quantizing and entropy-coding the wavelet coefficients, it uses only part of the coefficients for reconstruction and sets the rest to zero.

Figure 5 shows a definition of the evaluation function. It is an idealized version of a transform coder: Instead of quantizing and entropy-coding the wavelet coefficients, it uses only a certain percentage of the coefficients for reconstruction and sets the rest to zero. This approach is much less expensive and allows choosing the compression ratio exactly, which means that the resulting distortion can be used directly as a fitness measure. Villasenor et al. [24] have used a similar but even simpler method to evaluate wavelets with good results.

Figure 6 compares the performance predicted by the evaluation function to the actual performance in a transform coder, using wavelets tested during the experiments reported in the next section. The figure shows that the prediction

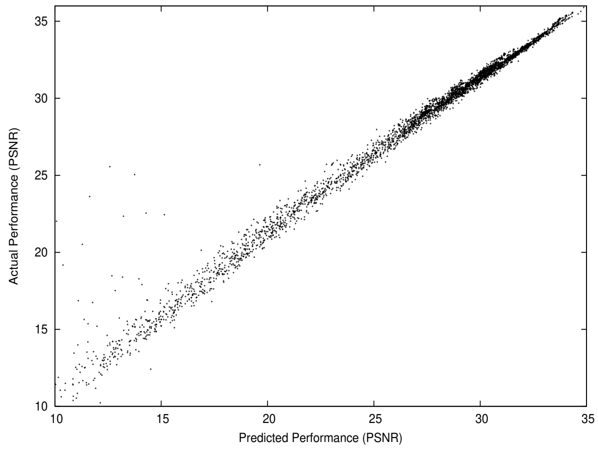


Figure 6: The relation between the performance predicted by the evaluation function (figure 5), and real-world performance in a transform coder. Each point represents a wavelet encountered in the experiments reported in section 5.

of the real-world performance is accurate, although in a few cases, the evaluation function underestimates the actual performance.

5. EXPERIMENTS

In this section, the wavelet evolution method described in the last section is applied to the compression of fingerprint images like the ones shown in figure 7.

5.1 Task and data

Fingerprint compression is both a popular research problem in image compression and an important application in its own right. The FBI alone has over 200 million fingerprint cards on file, occupying an acre of filing cabinets[2]. Recently, they have begun to store fingerprints electronically. Between 30,000 and 50,000 new cards are digitized and compressed every day, using a wavelet-based image coder and the so called 'FBI wavelet', considered to be the best known wavelet for fingerprint compression [13]. The FBI wavelet (shown in figure 1) therefore provides a challenging benchmark with which to compare the evolved wavelets.

For the experiments reported in this section, the first set of fingerprints from the FVC2000 finger print verification competition [19] was used. The data set contains 80 black-and-white images acquired electronically using an optical sensor ("Secure Desktop Scanner" by KeyTronic). The size of each image is 300 by 300 pixels, at 500 dpi resolution.

5.2 Methodology

Cross-Validation

The algorithm was evaluated using leave-one-out cross validation on the 80 available images, i.e. each of the images was used once as a test image, and 79 times as part of the training set. Each of the 80 runs took approximately 45 minutes on a 3GHz Xeon processor.

Figure 7: The FBI digitizes and compresses between 30,000 and 50,000 new fingerprint cards every day. The images shown are part of the data set used in the FVC2000 fingerprint verification competition [19], and were also used in the reported experiments.

Parameters

Preliminary experiments were conducted to determine the best parameter settings. The algorithm turned out to be very robust; similar results were obtained for a wide range of parameters.

The following parameters were used for the reported results: The population size was 150 for each sub-population, which means that 1500 evaluations took place in each generation. The algorithm evolved 7 sub-populations in parallel, each of which contained lifting steps of length 4. The mutation rate was set to 0.4. The compression ratio was 16:1 both for fitness evaluation and the evaluation on the test image. The evolution ran for 500 generations each time.

Quality Analysis

After each generation, the best wavelet found so far was used to compress the test image. The image coder was the implementation of the Set Partitioning in Hierarchical Trees (SPIHT) algorithm [21] in J. E. Fowler's QccPack, an open-source library of state-of-the-art data compression routines [8]. The performance on the test set can therefore be regarded as an accurate measure of real-world performance.

The following analysis relies on error images and peak signal-to-noise ratio (PSNR) as measures of image quality.

The PSNR is a simple logarithmic measure for the difference between two images. It is commonly used in image compression as a quantitative measure for the compression error introduced by a compression algorithm. A PSNR of 30 decibel is commonly regarded as a reasonable lower quality limit, and a PSNR above 50dB is regarded as "visually perfect", i.e. the compressed image is visually indistinguishable from the original. A difference of 0.4–0.5 dB between two algorithms is usually visible.

Error images are obtained by subtracting the compressed image pixel-by-pixel from the original. Error images are necessary because subtle differences in quality between compressed images are very hard to judge visually. In print, even substantial quality improvements are unlikely to be clearly visible. Separating the error from the image data makes comparisons much easier.

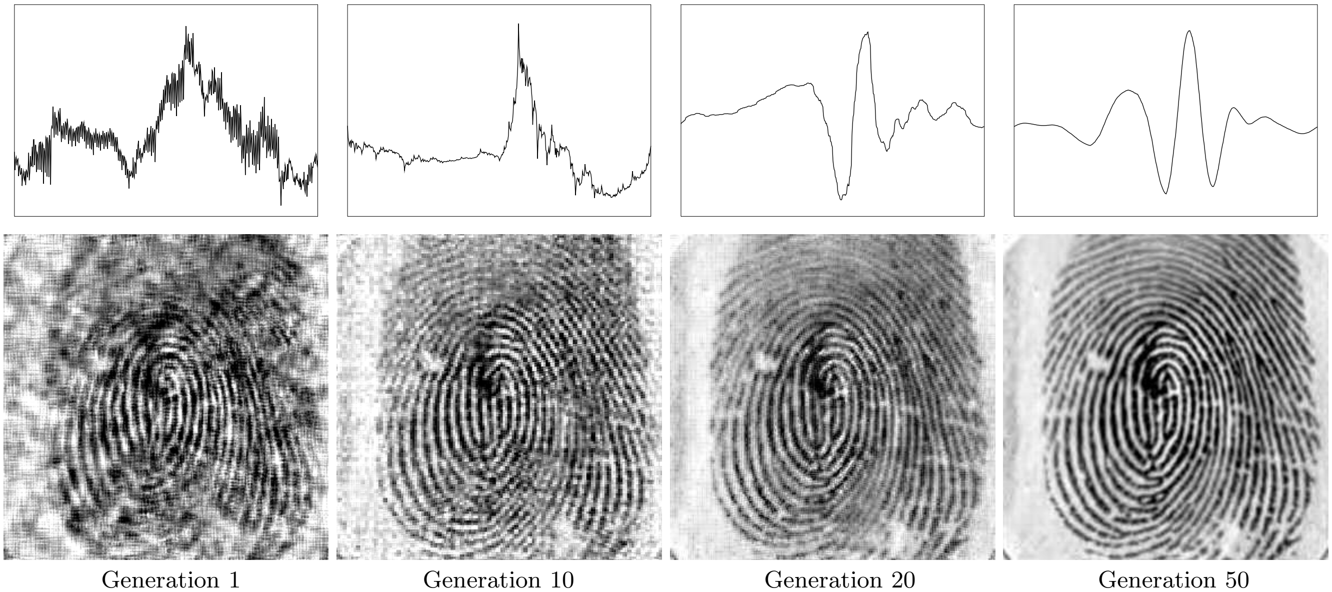


Figure 8: The progress of evolution during a typical run is shown at generations 1, 10, 20 and 50. The top row shows the winner wavelets, and the bottom rows shows the resulting compressed test image at 16:1. The first generation produced a more or less random wavelet that performs poorly. Over the next generations, both image quality and the smoothness of the wavelets increase sharply. The performance keeps increasing after generation 50 (figure 9), although the differences are less obvious.

5.3 Results

Figure 8 illustrates the progress of evolution during a typical run of the algorithm. The winner wavelets and the resulting compressed test images are shown after generations 1, 10, 20, and 50. The winner of the first generation is highly discontinuous and performs poorly. As the wavelets adapt to the structure of the data, they become smoother, and the image quality increases sharply. The performance keeps improving well beyond generation 50, although the differences in image quality are less obvious.

Figure 9a shows the learning curve of the algorithm, i.e. the performance on the test images, averaged over all 80 runs of the algorithm. The horizontal lines show the average performance of the FBI wavelet and a baseline JPEG coder. The curve shows that the evolved wavelets achieve an average improvement of 0.75dB over the FBI wavelet. The FBI wavelet in turn outperforms JPEG by approximately 2.5dB. Alternatively, a 0.75dB quality improvement would translate into a 15-20% decrease in space requirements for the same image quality.

Figure 9b compares the evolved wavelets to the FBI wavelet directly. The dotted line shows the lower limit of the 95% confidence interval. At 30 generations, the evolved wavelets perform the same on average as the FBI wavelet. By generation 40, there is a 95% probability of finding a better wavelet than the FBI wavelet. After 500 generations, the evolved wavelets are 95% certain to outperform the FBI wavelet by more than 0.45dB. At this point, the evolved wavelet is significantly better with $p < 10^{-4}$, according to a paired t-test.

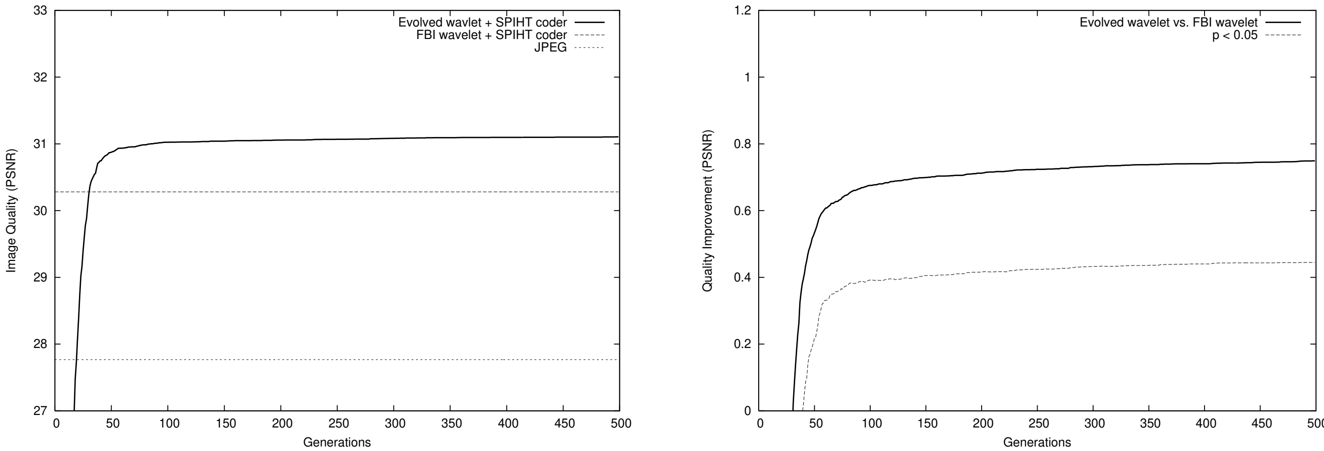
Figure 10 gives a visual impression of the increase in image quality achieved by the evolved wavelets. The top row shows a detail of the leftmost fingerprint in figure 7, compressed at 16:1 using the wavelet found by the GA, the FBI wavelet, and a baseline JPEG coder. Both wavelets perform much better than the JPEG coder: The blocking artifacts that often accompany JPEG-encoded images are clearly visible. The difference between the two wavelets is clearer in the bottom row, which shows only the error introduced by each compression method. The error image of the evolved wavelet is dimmer than that of the FBI wavelet, indicating a more accurate reconstruction of the original image. The constrast of the error images was increased uniformly to enhance the visibility of the non-zero differences.

6. DISCUSSION AND FUTURE WORK

The results presented in the previous section show that evolving wavelets for classes of images can considerably improve the performance of an image coder. The evolved wavelets substantially outperformed the hand-optimized FBI wavelet in every single run of the algorithm.

Many other image classes, including medical images, structural drawings, digitized documents, and satellite images have the characteristics that would make it useful to evolve specialized wavelets for them: They are often stored in large databases of similar images, they do not have the same statistical structure as photographs, and they have regularities of their own that standard wavelets cannot fully exploit. The results presented in this paper suggest that similar improvements should be possible in these cases. Other applications of the same algorithm, like lossless image compression, compression of volumetric data or wavelet-based multi-grid solvers for partial differential equations, are also possible in the future.

The algorithm itself could also be extended to evolve more powerful classes of wavelets. The design of non-separable and nonlinear wavelet transforms has received much attention in the literature recently (e.g. [10, 4, 23]). The algo-



(a) Learning curve.

(b) Evolved vs. FBI wavelet.

Figure 9: Learning curve of the wavelet evolution algorithm. In each generation, the best wavelet found was used in a state-of-the-art image coder. Plot (a) shows the resulting image quality on the test image, averaged over 80 runs. The horizontal lines show the performance of the FBI wavelet and a baseline JPEG coder. Plot (b) shows the average quality improvement over the FBI wavelet. The dotted line is the lower limit of the 95% confidence interval. After 500 generations, the evolved wavelets are 95% certain to outperform the FBI wavelet by at least 0.45dB, a difference usually visible to the naked eye.

rithm used in this paper could be adapted to evolve non-separable and nonlinear wavelets without major changes.

7. CONCLUSIONS

In this paper, a coevolutionary GA was used to evolve specialized wavelets for image compression, using fingerprint images as a test domain. The evolved wavelets consistently outperform the wavelet used by the FBI in this task. These results show that evolving wavelets adapted to specific image classes can significantly increase the compression performance of an image coder. They also demonstrate that evolutionary discovery can outperform significant human design effort in an important task.

8. ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation under grant IIS-0083776 and by the Texas Higher Education Coordinating Board under grant ARP-003658-476-2001.

9. REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1992.

[2] C. M. Brislawn. The fbi fingerprint image compression standard. http://www.c3.lanl.gov/~brislawn/FBI/FBI.html.

[3] R. Claypoole, R. Braniuk, and R. Nowak. Adaptive wavelet transforms via lifting. *Transactions of the International Conference on Acoustics, Speech and Signal Processing*, pages 1513–1516, May 1998.

[4] R. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *IEEE Transactions on Image Processing*, 12(12):1449 – 1459, December 2003.

[5] R. Coifman and V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38:713–718, 1992.

[6] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):245–267, 1998.

[7] G. Davis and A. Nosratinia. Wavelet-based image coding: An overview. *Applied and Computational Control, Signals and Circuits*, 1(1), 1998.

[8] J. E. Fowler. Qccpack: An open-source software library for quantization, compression, and coding. In e. A. G. Tescher, editor, *Applications of Digital Image Processing XIII*, volume Proc. SPIE 4115, pages 294–301, August 2000.

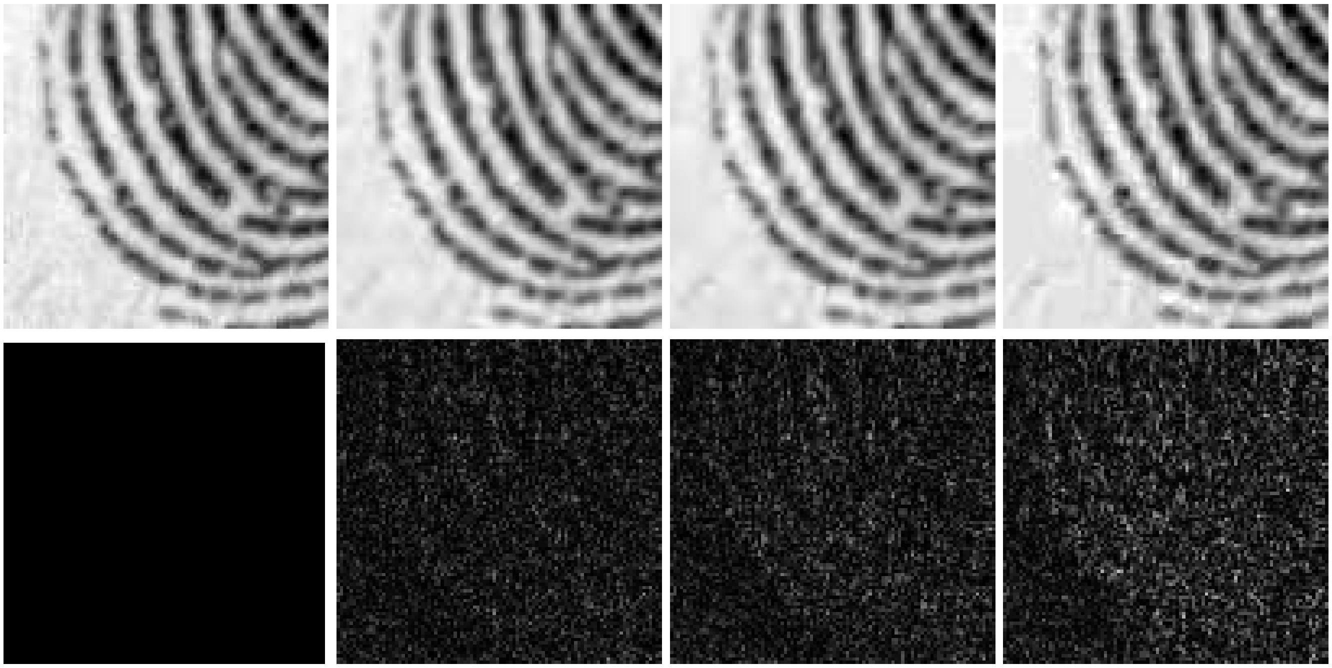
[9] F. Gomez and R. Miikkulainen. Solving non-markovian control tasks with neuroevolution. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1356–1361, San Francisco, CA, 1999.

[10] A. Gouze, M. Antonini, M. Barlaud, and B. Macq. Design of signal-adapted multidimensional lifting scheme for lossy coding. *IEEE Transactions on Image Processing*, 13(12):1589 – 1603, December 2004.

[11] U. Grasemann and R. Miikkulainen. Evolving wavelets using a coevolutionary genetic algorithm and lifting. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume II, pages 969–980, New York, NY, 2004. Springer.

[12] Y. Hill, S. O'Keefe, and D. Thiel. An investigation of wavelet design using genetic algorithms. In *Microelectronic Engeneering Research Conference*, 2001.

[13] T. Hopper, C. M. Brislawn, and J. N. Bradley. *WSQ Gray-scale Fingerprint Image Compression Specification*. Federal Bureau of Investigation, February 1993. Document No. IAFIS-IC-0110.



(a) Original (b) Evolved (c) FBI Wavelet (d) JPEG

Figure 10: (a) A detail of the left fingerprint in figure 7, compressed at 16:1 with an evolved wavelet (b), the FBI wavelet (c), and a baseline JPEG coder (d). The bottom row shows the error introduced by each compression method. The error image of the evolved wavelet is dimmer than that of the FBI wavelet, reflecting the lower overall distortion.

[14] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.

[15] I. JTC1/SC29/WG1. JPEG 2000 - lossless and lossy compression of continuous-tone and bi-level still images. Part I: Minimum decoder. Final committee draft, version 1.0., March 2000.

[16] M. M. Lankhorst and M. D. van der Laan. Wavelet-based signal approximation with genetic algorithms. In *Evolutionary Programming*, pages 237–255, 1995.

[17] R. L. Lippke. The disenfranchisement of felons. *Law and Philosophy*, 20:553 – 580, Nov 2001.

[18] C. Liu and H. Wechsler. Face recognition using evolutionary pursuit. In *Proceedings of the Fifth European Conference on Computer Vision*, Freiburg, Germany, 1998.

[19] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, New York, June 2003.

[20] D. Monro and B. Sherlock. Space-frequency balance in biorthogonal wavelets. *Transactions of the IEEE Int. Conf. on Image Processing*, 1:624–627, 1997.

[21] A. Said and W. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, June 1996.

[22] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.

[23] W. Sweldens. The lifting scheme: A construction of second-generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[24] J. Villasenor, B. Belzer, and J. Lia. Wavelet filter evaluation for image compression. *IEEE Transactions on Image Processing*, 2:1053–1060, August 1995.

[25] G. Wallace. The jpeg still picture compression standard. *IEEE TCE*, 38, 1992.

[26] M. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A. K. Peters, Wellesley, MA, 1994.