

# Evolutionary design and optimization of Wavelet Transforms for image compression in embedded systems

Rubén Salvador, Félix Moreno, Teresa Riesgo  
Centre of Industrial Electronics  
Universidad Politécnica de Madrid  
José Gutiérrez Abascal, 2  
28006, Madrid, Spain  
Email: ruben.salvador@upm.es

Lukáš Sekanina  
Faculty of Information Technology  
Brno University of Technology  
Božetěchova 2  
612 66 Brno, Czech Republic  
Email: sekanina@fit.vutbr.cz

## Abstract

*This paper describes the initial studies of an Evolution Strategy aimed at implementation on embedded systems for the evolution of Wavelet Transforms for image compression. Previous works in the literature have already been proved useful for this application, but they are highly computationally intensive. Therefore, the work described here, deals with the simplifications made to those algorithms to reduce their computing requirements. Several optimizations have been done in the evaluation phase and in the EA operators. The results presented show how the proposed algorithm cut outs still allow for good results to be achieved, while effectively reducing the computing requirements.*

## 1. Introduction

Embedded systems engineering faces an enormous challenge derived from the ever-increasing demand of more versatile electronic systems in each generation. At the same time, a great capacity of adaptation to very diverse operating environments is required.

One of such type of systems, mobile devices (in general, ubiquitous computing devices) in this new multimedia-oriented world, demand high performance signal processing capabilities, such as still image and video processing. For these reasons, compression standards require more capability of signal —image, video, etc.— compression.

Compression standards, such as JPEG2000 [1] are able to deal with an increased compression capability when compared to previous standards while keeping the transmission size. JPEG 2000 is based on the Discrete Wavelet Transform (DWT) [2], [3] in contrast

to the previous JPEG standard based on the Discrete Cosine Transform (DCT). Its main drawback —if it can be considered as so— is that the transform performance, from the compression point of view, which is determined by the wavelet used, will deteriorate if a different type of image than the one the wavelet is adapted to is used. The accepted standard wavelet considered to be the state of the art in compression of photographic images is the hand-designed D9/7 Cohen-Daubechies-Feauveau (9/7-CDF or also D9/7) [2]. Therefore, though being better in terms of coding capability when compared to JPEG, its performance highly depends on this chosen wavelet, so good performance is expected on the type of images the wavelet is adapted to.

This work deals with finding new wavelets for specific type of images by means of evolutionary algorithms (EA), specifically using Evolution Strategies (ES) [4]. The intended deployment platform is an FPGA embedded system. Therefore, a relatively low computing power is available what will, undoubtedly, affect the performance of the evolutionary search, but provides the system with adaptation capabilities, so that image compression performance can be adjusted for the specific type of images the encoder is dealing with. Therefore, a hardware oriented algorithm has to be developed to assure the feasibility of the implementation. For these reasons, issues such as the use of complex evolutionary operators and a fixed point implementation and validation of the algorithm have to be addressed.

The paper is structured as follows. Firstly, a brief overview of the Wavelet Transform is given in Section 2, along with a compilation of previous published works on the topic on Section 3. The proposed al-

gorithm is presented in Section 4. Section 5 presents the experimental setup, the results obtained and some additional comments on the future work. The paper is concluded in Section 6.

## 2. Overview of the Wavelet Transform

The DWT is a multiresolution analysis (MRA) tool widely used in signal processing due to its joint time-frequency signal analysis characteristics, helping in the analysis of the frequency content of a signal at different resolutions.

Similarly to the Fourier Transform, which decomposes a signal as a sum of sines at different frequencies, the Wavelet Transform represents the signal as a superposition of different *wavelets*, which are localized waves that have their energy concentrated within a variable time window. It is very similar to the Short Time Fourier Transform (STFT), in which the signal is multiplied by a constant width window function which is shifted along the signal computing the Fourier Transform of each sub-signal. In the case of the Wavelet Transform, the signal is decomposed in scaled and dilated versions of a wavelet function, called the *mother wavelet*. The main advantage of the wavelet transform is the variable width of its *window function*, the *wavelet*, that changes for each spectral component to be computed, overcoming the inability of the STFT of giving good time resolution and poor frequency resolution at high frequencies and good frequency resolution but poor time resolution for low frequencies.

The DWT helps to concentrate the signal energy into fewer coefficients to increase the degree of compression when the data is encoded. The energy of the input signal is redistributed into a low resolution trend sub-signal (scaling coefficients) and high resolution sub-signals (wavelet coefficients; horizontal, vertical and diagonal sub-signals for image transforms). If the wavelet chosen for the transform is suited for the type of image being analyzed, most of the information of the signal will be kept in the trend sub-signal, while the wavelet coefficients (high frequency details) will have a very low value. For this reason, the DWT can reduce the number of bits required to represent the input data.

For a general introduction to wavelet based multiresolution analysis see [5]. Using the algorithm called *Fast Wavelet Transform* (FWT), the wavelet representation can be computed by recursively filtering the input data with a pair of high-pass and low-pass digital filters and downsampling the results by a factor of two [3]. This scheme, known in signal processing as *subband filtering*, is shown on Figure 1, where  $g$  and  $h$  are

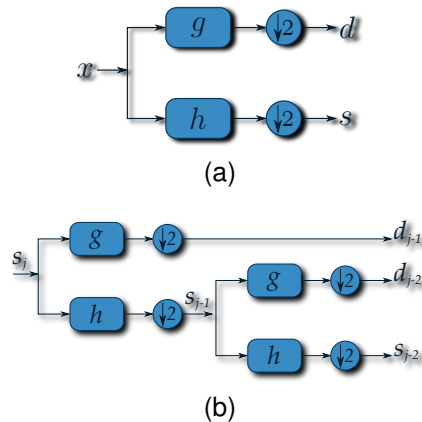


Figure 1. Fast Wavelet Transform: (a) shows the filter bank computation scheme. (b) is an example of a two level wavelet based MRA.

high-pass and low-pass FIR filters respectively, and  $d$  and  $s$  the wavelet and scaling coefficients. For multiresolution analysis, the scaling coefficients resulting from the first FWT computation ( $s_{j-1}$ ) are used as input data for the next decomposition level, yielding, again, a set of wavelet ( $d_{j-2}$ ) and scaling ( $s_{j-2}$ ) coefficients, as shown on Figure 1b. Therefore, the wavelet representation of  $s_j$  is given by the set of coefficients  $\{s_{j-2}, d_{j-2}, d_{j-1}\}$ . The D9/7 wavelet gets its name because their  $h$  and  $g$  filters have 9 and 7 coefficients respectively. There also exists a different notation for the transform coefficients  $\{s_{j-i}, d_{j-i}\}$ ; for a 2 level image decomposition it is  $\{LL, LH, HL, HH\}$  where  $L$  stands for low pass and  $H$  for high pass coefficients respectively.

The FWT algorithm was improved by the method known as *Lifting scheme* (LS), introduced by Sweldens [6]. The main advantages of the LS are summarized below:

- Faster (and easier) implementation of the wavelet transform.
- Fully in-place calculation. No auxiliary memory is needed.
- Easy inverse transform by *reversing* operations order and making a simple swap of  $+$ ,  $-$  signs.
- Simple extension to integer to integer transforms.
- Custom design the wavelet for particular applications.

Besides, the LS, which does not rely on the Fourier Transform for its definition and application, has given rise to the so called *Second Generation Wavelets* [7]. Some of the main advantages of these new wavelets are listed below:

- Transform signals with finite or arbitrary length

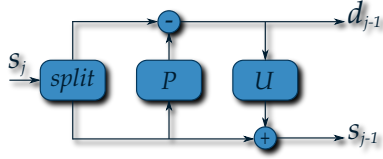


Figure 2. Lifting scheme

without introducing artifacts at the boundaries.

- Transform irregularly sampled data.
- Can be used on arbitrary geometries.

The basic LS, shown on Figure 2, consists of three stages: *Split*, *Predict* and *Update*. The main idea consists on exploiting the correlation structure of the input data to obtain a more compact representation of the signal [8].

*Split* stage divides the input data into two smaller subsets,  $s_{j-1}$  and  $d_{j-1}$ , usually, the even and odd samples. It is also called the *Lazy Wavelet*.

To obtain a more compact representation of the input data, the  $s_{j-1}$  subset is used to *predict* the  $d_{j-1}$  subset, called the wavelet subset, and which is based on the correlation of the original data. The difference between the prediction and the actual samples is stored, also, as  $d_{j-1}$ , overwriting its original value. If the prediction operator  $P$  is reasonably well designed, the difference will be very close to 0, so the two subsets  $s_{j-1}$  and  $d_{j-1}$  yield a more compact representation of the original data set  $s_j$ .

In most cases, it is interesting to maintain some properties of the original signal after the transform, such as the mean value. For this reason, the LS proposes a third stage that not only reuses the computations already done in the previous stages, but that also defines an easily invertible scheme. This is accomplished by *updating* the  $s_{j-1}$  subset with the already computed wavelet set  $d_{j-1}$ .

This scheme can be iterated up to  $n$  levels, as in the case of the filter bank computation. In this case, an original input data set  $s_0$  has been replaced with the wavelet representation  $\{s_{-n}, d_{-n}, \dots, d_{-1}\}$ . Below, the FWT LS algorithm is shown:

```

for  $j \leftarrow 1, n$  do
   $\{s_j, d_j\} \leftarrow Split(s_{j+1})$ 
   $d_j = d_j - P(s_j)$ 
   $s_j = s_j + U(d_j)$ 
end for

```

### 3. Previous work on evolutionary wavelets design

Research on adaptive wavelets has been active during the last two decades. Basically, dictionary-based methods, in some cases combined with EA, as well as several stochastic optimization techniques like simulated annealing, or even using the LS technique, have been reported.

The work described on this paper gets its original idea of [9] by Grasmann and Miikkulainen. In their work, the authors proposed the original idea of combining the lifting technique with EA for designing wavelets. As it is drawn from [6], [7] the LS is really well suited for the task of using an EA to encode wavelets, since any random combination of lifting steps will encode a valid wavelet, what guarantees perfect reconstruction.

Table 1 shows the most remarkable and up to date published results in the design of wavelet transforms by means of Evolutionary Computation (EC), and Table 2 the parameters settings for each reported work. It should be noted that in the cases of MRA, the coefficients evolved each at level were different, since the authors reported better results with this scheme, with the exception of [9].

Grasmann and Miikkulainen method is based on a coevolutionary GA that encodes wavelets as a sequence of lifting steps, which is based on the Enforce Sub-Populations (ESP) neuroevolution, that was reported by the same authors in [14]. The evaluation run making combinations of one individual (lifting step) from each sub-population until each individual had been evaluated an average of 10 times. Since this is a highly time consuming evaluation, in order to save time on the evaluation of the resulting wavelet, only a certain percentage of the largest coefficients is used for reconstruction, setting the rest to zero. Exactly, a 16:1 compression ratio was used, so 6.25% of the coefficients are kept for reconstruction. A comparison between the idealized evaluation function and the performance on a real transform-coder is shown on their work. Peak signal-to-noise-ratio (PSNR) was the fitness figure used as a quality measure after performing the inverse transform. The fitness for each lifting step was accumulated each time it was used.

The 80 images used for the algorithm evaluation were obtained from the first set of images of the FVC2000 fingerprint verification competition [15]. Images were black and white, sized 300x300 pixels at 500 dpi resolution. A cross validation strategy was used, where each image was used once as a test image and 79 times as part of the training set. Therefore, the

Table 1. State of the Art in evolutionary wavelets design.

Ref.	EA	Seed	Conditions	Image set	Result (improvement)
[10]	GA	D4 mutations	MRA (3). 64:1 Q <sup>a</sup>	Photographs	0.60 dB (MSE)
[11]	GA	D9/7 mutations	MRA (4). 16:1 T <sup>b</sup>	Fingerprints Satellite	0.76 dB (MSE) <b>1.79 dB</b> (MSE)
[12]	CMA-ES <sup>c</sup>	D9/7 mutations	64:1 Q	Fingerprints Photographs	<b>3.00 dB</b> <b>2.39 dB</b>
[13]	CMA-ES	0.2	MRA (3). 64:1 Q	Fingerprints	0.54 dB (MSE)
[9]	Coevolutionary GA	Random Gaussian	MRA. 16:1 T	Fingerprints	0,45 dB (PSNR)

<sup>a</sup>Quantization <sup>b</sup>Thresholding <sup>c</sup>Covariance Matrix Adaptation-Evolution Strategy

evaluation consisted of 80 runs, each of which took approximately 45 minutes on a 3 GHz Xeon processor. The results obtained in this work outperformed the considered state-of-the-art wavelet for fingerprint image compression, the FBI standard based on the D9/7 wavelet, in 0.45 dB.

Works reported by Babb, Moore et. al. [10], [11], [12], [13], [16], [17], [18], [19], can be considered the current state of the art in the use of EC for image transform design. The milestones followed in their research, with references to their first published works, are summarized on the next list.

- 1) Evolve the inverse transform for digital photographs under conditions subject to quantization [16].
- 2) Evolve *matched* forward and inverse transform pairs [17].
- 3) Evolve coefficients for three and four level MRA transforms [18].
- 4) Evolve a different set of coefficients for each of level of MRA transforms [10].

As stated by the authors, Babb, Moore, et al., their algorithms are highly computationally intensive, so the training runs were done using supercomputing resources, available through the use of the Arctic Region Supercomputer Center (ARSC) in Fairbanks, Alaska. Although the work by Grasemann and Miikkulainen was done on an *accessible* computer, both training times and computing resources needed in both cases, show the complexity of the algorithms developed. These approaches are highly unfeasible for an implementation as a hardware embedded system.

#### 4. Proposed Evolutionary Algorithm

In order to find a viable hardware implementation the whole process of evolution has to be down-scaled in complexity. This implies changing not only the parameters of evolution, but the EA itself. Although the hardware architecture of an FPGA may be considered the paradigm of parallelism, the complexity of such

Table 2. Parameter settings in reported work.

Ref.	Parameters	Platform
[10]	$G^a= 500, M^b= 2000, N^c= 48$	ARSC <sup>d</sup>
[11]	$G = 15000, M = 800, N = 128$	ARSC
[12]	$G = ?^e, M = ?, N = 16$	ARSC
[13]	$G = ?, M = ?, N = 96$	ARSC
[9]	$G = 500, M = 150(7)^f, N = 4 + 1^g$	ARSC

<sup>a</sup>Generations <sup>b</sup>Population size

<sup>c</sup>Individuals length (floating point coefficients)

<sup>d</sup>Arctic Region Supercomputer Center <sup>e</sup>Unknown

<sup>f</sup>Parallel sub-populations <sup>g</sup>Integer for filter index

an implementation is much higher than the software choice. So, a trade-off between scalability, use of hardware resources and difficulty (complexity) of the implementation has to be taken into account when setting system performance requirements. Anyhow, a prototype implementation is needed to evaluate the resource requirements before a final decision is taken. Therefore, the work presented here is an initial study towards a future hardware implementation of the system.

The work done by Grasemann and Miikkulainen [9] was the starting point of this work. As shown on Section 3, Babb, Moore, et al. have also been contributing to this field of research, but their last contributions [12], [13] were not known by the time this work began. In the same direction as them, since the target evolution parameters were floating point numbers representing filter coefficients, an Evolution Strategy seemed to be also the most suited algorithm. But, differently from their work, a simpler ES was chosen, so that the system requirements trade-off was kept. In the next paragraphs, the decisions made for designing a simpler algorithm are presented.

First of all, evolving parallel populations as does the analyzed work for an intended hardware implementation on an FPGA, would be very expensive in terms of computing resources, so the coevolutionary genetic

algorithm used in [9] is discarded at this stage of the work.

In [9], an ideal version of a transform coder was used as part of the evaluation phase. This is due to the fact that doing a complete compression would turn to be an unsustainable amount of computing time. Although the evaluation method reported by Grasmann and Miiikkulainen was highly time saving, for the scope of this work, it has been improved in terms of computing effort. Therefore, this work goes even further, zeroing completely all wavelet coefficients  $d_j$  and keeping only the trend level  $s_j$  of the transform from the last iteration of the algorithm, *without previously ordering and zeroing* the resulting coefficients at each iteration level. While being a very radical proposal, it is also very radical in saving computing time, area and power in the FPGA, freeing up resources for other tasks, as well as coherent enough to expect good results. Therefore, the evaluation of the population individuals is done by computing the PSNR after setting entire bands of high-pass coefficients to 0. Two decomposition levels are done and only the LL coefficients are kept before reconstructing the image and measuring the resulting error. For 2 levels of decomposition, this is an idealized 16:1 compression ratio.

CMA-ES [20], used as the EA in [12], [13], uses an over-complex mutation strategy for a hardware implementation. In this work, a much simpler mutation operator has been used. As proposed in [4] uncorrelated mutations of one endogenous strategy parameter have been used for the variation operator, which fulfills the ideas of self-adaptation. The mutation operator in ES is based on a normal (Gaussian) distribution, with mean  $\xi$  and standard deviation  $\sigma$ . Mutations are done by adding some  $\Delta x_i$  to each parameter  $x_i$  encoded in the chromosome. Besides, the particular feature of mutation in ES is that the step sizes are also included in the chromosomes, undergoing variation and selection themselves. To sum up, the algorithm used in this work proposes:

- 1) Evolving one single population.
- 2) Use of uncorrelated mutations with one step size.
- 3) Evolution of one single set of coefficients for all MRA levels.
- 4) Ideal evaluation of the reconstructed image.

## 5. Experimental setup, results and discussion

### 5.1. Experimental setup

This paper deals mainly with the design modeling and simulation of the algorithm in a high level

language prior to its hardware implementation, which is being carried out at present. For this initial study, Python computing language [21], together with its numerical and scientific extensions, NumPy and Scipy [22], as well as the plotting library Matplotlib [23], was chosen for this purpose. The computer used was a laptop containing an Intel Core™2 Duo processor at 2 GHz running a Debian GNU/Linux 64 bits operating system. In order to validate the proposed algorithm, several tests for different parameters combinations were carried out to determine if the simplified algorithm works and to find suitable settings for these parameters. Choices are based on the recommendations given in [4], [24].

Standard **representation** of the individuals is composed of a set of *object parameters* to be optimized, and a single *strategy parameter*, which determines the degree of perturbation of the mutation operator:

$$\langle x_1, \dots, x_n, \sigma \rangle$$

with  $x_i$  being floating point values representing the coefficients of the predict and update stages.

The encoding of each wavelet individual is of the form:

$$\langle P_1, U_1, P_2, U_2, P_3, U_3, k1, k2 \rangle$$

where each  $P_i, U_i$  is made up of 4 coefficients and  $k_i$  are single coefficients. Therefore, the total length of each chromosome is  $n = 26$  floating point values. D9/7 wavelet uses  $\langle P_1, U_1, P_2, U_2, k1, k2 \rangle$ .

The **mutation** operator is an *uncorrelated mutation with one step size*,  $\sigma$ . The formulas for the mutation mechanism are:

$$\sigma' = \sigma \cdot \exp^{\tau \cdot N(0,1)} \quad (1)$$

$$x'_i = x_i + \sigma' \cdot N_i(0,1) \quad (2)$$

where  $N(0,1)$  is a draw from the standard normal distribution and  $N_i(0,1)$  a separate draw from the standard normal distribution for each variable  $i$  (for each object parameter). The parameter  $\tau$  resembles the so called *learning rate* of neural networks, and it is proportional to the square root of the object variable length  $n$ :

$$\tau \propto 1/\sqrt{n}$$

As other published works, the **fitness function** used to evaluate the offspring individuals is also the PSNR:

$$MSE = \frac{1}{RC} \sum_{i=0}^{R-1} \sum_{j=0}^{C-1} |I(i,j) - K(i,j)|^2 \quad (3)$$

$$PSNR = 10 \log_{10} \left( \frac{I_{\max}}{MSE} \right) \quad (4)$$

Table 3.  $(\mu/\rho, \lambda)$ -ES parameters for evolving wavelets.

Parameter / Operator	Value
<b>Representation</b>	Floating point coefficients $n = 26$
<b>Recombination</b>	Intermediate, global $\rho = 5$
<b>Mutation</b>	Self-adaptive, one-step size $\sigma = \{0.1, \dots, 1.5\}$
<b>Parent population size</b>	$\mu = 10$
<b>Parent selection</b>	Uniform random
<b>Offspring population size</b>	$\lambda = 70$
<b>Seed for initial population</b>	Random and D9/7
<b>Survivor selection</b>	Comma $(\mu, \lambda)$
<b>Learning rate</b>	$\tau \propto 1/\sqrt{\alpha n}, \alpha = \{1, 2\}$

where  $R, C$  are the rows and columns of the image,  $I, K$  the original and transformed images respectively and  $I_{\max}$  the maximum possible value of a pixel. For  $B$  bpp  $I_{\max} = 2^{B-1}$ .

For the *survivor selection* after creating  $\lambda$  offspring and calculating their fitness, the best  $\mu$  individuals are selected deterministically for the new parent population based on their rank fitness value. The selection scheme used,  $(\mu, \lambda)$ , is called *comma selection*, with a selection pool filled with the  $\lambda$  offspring individuals. It is generally preferred in ES over the *plus selection*  $(\mu + \lambda)$ , where the selection pool consists of the  $\mu + \lambda$  individuals of the parent and offspring populations, for being, in principle, able to leave (small) local optima and not letting survive misfit strategy parameters. Therefore, no *elitism* is allowed.

The *recombination* scheme chosen is *intermediate recombination*, where the alleles of the selected parents are averaged. The selection of these  $\rho$  recombinants is done randomly from the parent population. If  $\rho = 2$  the scheme is called local recombination and if  $\rho > 2$ , global recombination. Table 3 shows a summary of the chosen parameters.

The experiments reported in this work have also used, as in [9], the first set of 80 images of the FVC2000 fingerprint verification competition [15]. One random image was used for training and the other 79 for testing purposes. The evolution run for 1000 generations, taking around 35 minutes to complete, much less than the analyzed works.

## 5.2. Results

All the results obtained are compared with the D9/7 transform evaluated with the proposed method, whose average measured PSNR for the 79 test images is 29.60

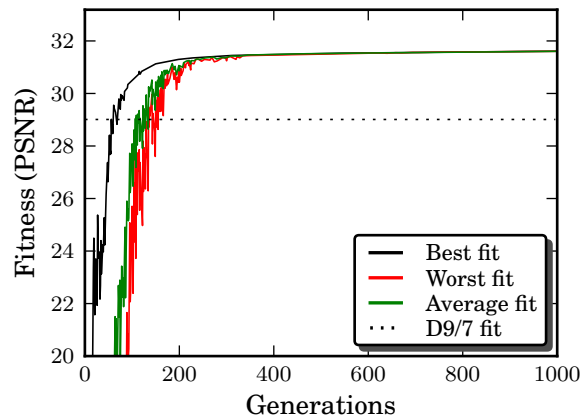


Figure 3. Result of the best evolution run

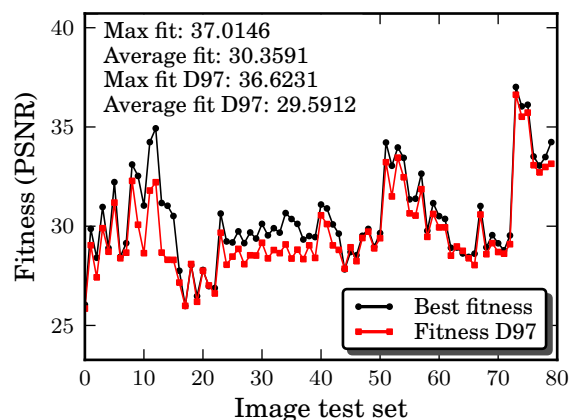


Figure 4. Tests of the best evolved wavelet. The best individual (and D9/7 for comparison) is exercised for each of the 79 images of the test set

dB. Table 4 shows the results of different evolution runs for different settings of the parameters (total 65 runs to build the whole Table). The best evolved wavelet, marked in bold, outperforms the D9/7 by an average of 0.76 dB, improving some of the published results shown in Table 1.

Figure 3 shows how the algorithm behaves during a typical run. As it can be seen, around generation 300 the performance of the evolved wavelet is already similar to the D9/7. In Figure 4 the best evolved wavelet is compared with the D9/7 for the whole image test set, showing how the algorithm was able to evolve a solution that outperforms the standard D9/7.

## 5.3. Discussion and future work

The results obtained are comparable to (though not outperform) the state of the art. This shows the feasibil-

Table 4. Results for the different parameter settings.

Seed	Seed Mask <sup>a</sup>	$\sigma(\Delta\sigma = 0.1)$	$\alpha$	G	Average Maximum <sup>bf</sup>	Average <sup>cf</sup>	Best Maximum <sup>df</sup>	Best Average <sup>ef</sup>
D9/7	False	{0.1, ..., 1.5}	2.0	1000	36.79	30.10	37.01	30.36
D9/7	True	{0.5, ..., 1.5}	1.0	1300	36.30	29.41	36.62	29.62
Random	–	{0.5, ..., 1.5}	2.0	1300	36.48	29.92	36.87	30.19
Random	–	{0.5, ..., 1.5}	1.0	1300	36.55	29.93	37.01	<b>30.44</b>
Random	–	{0.1, ..., 1.5}	2.0	1000	36.18	29.64	37.00	30.34

<sup>a</sup> True for 9/7 initialized wavelets with shorter chromosome (exact structure of the 9/7),  $\langle P_1, U_1, P_2, U_2, k_1, k_2 \rangle$

<sup>b</sup> Average of the maximum fitness for the 79 test images and for the variable sigma <sup>c</sup> Average of the average fitness for the 79 test images and for the variable sigma <sup>d</sup> Maximum of the best wavelet found in the whole run <sup>e</sup> Average of the best wavelet found in the whole run <sup>f</sup> All results in dB

ity of the proposal (for a very standard set of parameters) as a simplified algorithm to be implemented on an FPGA, where performance is sacrificed up to an extent. Further tests for a wider variation of all the parameters in the algorithm shall be done in order to improve it while keeping the required computing resources. Besides, different operators for the ES shall be tested, as for example, an even simpler mutation scheme, the *two-point rule* [25]. But special care should be taken here since it has been observed during the several runs of the algorithm that it is somehow prone to stagnation at local optima. Therefore, as suggested in the literature of ES, a boundary rule to avoid this behavior shall be observed. It is also worth to try a different approach for the recombination of both the strategy and object parameters, mixing intermediate with discrete recombination, as suggested in [24]. As for the evaluation phase, a different method will try to be developed to avoid the stagnation of the evolution. The methodology followed allows these modifications to be more easily tested than if directly implemented in hardware.

For the implementation of the algorithm in an FPGA, the EA has to be tested with fixed point binary arithmetic, more hardware friendly than floating point. Thanks to the LS, the Integer Wavelet Transform (IWT) [26] turns up as a good solution for wavelet transforms in embedded systems. However, floating point arithmetic is still around in the system, in the form of filter coefficients. But, although due to the very nature of the IWT perfect reconstruction is assured, arithmetic performance has to be observed when using fixed point operations, since reducing the bit length excessively may yield unacceptable results due to rounding errors. As shown on [27], [28], for 8 bits per pixel (bpp) integer inputs from an image, a fixed point fractional format of Q2.10 for the lifting coefficients and a bit length in between 10 and 13 bits for a 2 to 5 level MRA transform for the partial results is enough to keep a rate-distortion performance almost equal to

what is achieved with floating point arithmetic. This yields Multiply and Accumulate (MAC) units of 20-23 bits (10 bits fractional coefficients part + 10-13 bits for partial transform results).

## 6. Conclusion

In this paper, a relatively simple ES was developed to evolve wavelet transforms for image compression. Although the evolved wavelets do not outperform the State of the Art reported work, which use more complex and computationally intensive algorithms, still behave better in average than the D9/7 wavelet, used in JPEG2000 and in the FBI compression standard. These results open the possibility of further investigations for the implementation of the algorithm in an FPGA from the point of view of, both, hardware acceleration of evolution and an adaptive embedded system.

## Acknowledgment

This work was supported by the Spanish Ministry of Science and Research under the project DR.SIMON (Dynamic Reconfigurability for Scalability in Multimedia Oriented Networks) with number TEC2008-06486-C02-01.

Lukas Sekanina has been supported by MSMT under research program MSM0021630528 and by the grant of the Czech Science Foundation GP103/10/1517.

The main author would like to thank the support received from the Department of Computer Systems, Brno University of Technology, during his research stay as part of his PhD degree.

## References

- [1] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, 1st ed. Springer, Nov. 2001.
- [2] I. Daubechies, *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.

- [3] S. Mallat, *A Wavelet Tour of Signal Processing, Second Edition*, 2nd ed. Academic Press, Sep. 1999.
- [4] H. Beyer and H. Schwefel, "Evolution Strategies. A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, Mar. 2002.
- [5] B. Jawerth and W. Sweldens, "An overview of wavelet based multiresolution analyses," *SIAM Review*, vol. 36, no. 3, pp. 377–412, 1994.
- [6] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186 – 200, 1996.
- [7] —, "The lifting scheme: a construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, 1998.
- [8] —, "The lifting scheme: a new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, Eds., vol. 2569, no. 1. SPIE, 1995, pp. 68–79.
- [9] U. Grasmann and R. Miikkulainen, "Effective image compression using evolved wavelets," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. Washington DC, USA: ACM, 2005, pp. 1961–1968.
- [10] F. Moore and B. Babb, "Revolutionary image compression and reconstruction via evolutionary computation, part 2: multiresolution analysis transforms," in *Proceedings of the 6th WSEAS International Conference on Signal, Speech and Image Processing*. Lisbon, Portugal: World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 144–149.
- [11] B. Babb and F. Moore, "The best fingerprint compression standard yet," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 2911–2916.
- [12] B. Babb, F. Moore, M. Peterson, T. H. O'Donnell, M. Blowers, and K. L. Priddy, "Optimized satellite image compression and reconstruction via evolution strategies," in *Evolutionary and Bio-Inspired Computation: Theory and Applications III*, vol. 7347. Orlando, FL, USA: SPIE, May 2009, pp. 734700–10.
- [13] B. J. Babb, F. W. Moore, and M. R. Peterson, "Improved multiresolution analysis transforms for satellite image compression and reconstruction using evolution strategies," in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. Montreal, Quebec, Canada: ACM, 2009, pp. 2547–2552.
- [14] F. J. Gomez and R. Miikkulainen, "Solving non-markovian control tasks with neuroevolution," in *IJCAI '99: Proceedings of the 16th International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 1356–1361.
- [15] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "FVC2000: fingerprint verification competition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 3, pp. 402–412, Mar 2002.
- [16] F. Moore, P. Marshall, and E. Balster, "Evolved transforms for image reconstruction," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3, 2005, pp. 2310–2316 Vol. 3.
- [17] B. Babb, S. Becke, and F. Moore, "Evolving optimized matched forward and inverse transform pairs via genetic algorithms," in *Circuits and Systems, 2005. 48th Midwest Symposium on*, 2005, pp. 1055–1058 Vol. 2.
- [18] F. Moore, "A genetic algorithm for evolving multiresolution analysis transforms," in *WSEAS Transactions on Signal Processing*, vol. 1, 2005, pp. 97–104.
- [19] F. W. Moore and B. Babb, "A differential evolution algorithm for optimizing signal compression and reconstruction transforms," in *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*. Atlanta, GA, USA: ACM, 2008, pp. 1907–1912.
- [20] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a New Evolutionary Computation*, 2006, pp. 75–102.
- [21] G. V. Rossum, *The Python Language Reference Manual*. Network Theory Ltd., Sep. 2003.
- [22] T. E. Oliphant, "Python for scientific computing," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 10–20, 2007. [Online]. Available: <http://link.aip.org/link/?CSX/9/10/1>
- [23] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [24] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer, Oct. 2008.
- [25] H. Beyer, *The Theory of Evolution Strategies*, 1st ed. Springer, Apr. 2001.
- [26] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332 – 369, 1998.
- [27] M. Martina, G. Masera, G. Piccinini, and M. Zamboni, "A VLSI architecture for IWT (integer wavelet transform)," in *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on*, vol. 3, 2000, pp. 1174–1177 vol.3.
- [28] M. Grangetto, E. Magli, M. Martina, and G. Olmo, "Optimization and implementation of the integer wavelet transform for image coding," *Image Processing, IEEE Transactions on*, vol. 11, no. 6, pp. 596–604, 2002.