Best basis selection for image compression using cellular automata transforms.

Juan E. Paz

Center for Studies on Electronics and Information Technologies, Central University of Las Villas, Zip code: 54830, Santa Clara, Cuba.
jpaz@ceeti.uclv.edu.cu

Abstract. Cellular Automata Transform is an alternative tool for image and signal processing like *Fourier* or *Wavelet* transforms. From the big number of transformation bases that can be generated using this method is possible to find the one more accurate for the problem in question that when using previous transforms. The present paper deals with an efficient basis selection criterion measured though the entropy of the transformed coefficients in an application like image compression.

1 Introduction

Cellular Automata are dynamical systems in which space and time are discrete. They were presented for the first time by *von Neumann* and *Ulam* [1] and had been progressively used in modeling a great variety of dynamic systems in diverse applications [2]. Like former transforms as *Fourier* or *Wavelet*, the Cellular Automata Transform (CAT) finds a practical application in digital image processing like compression, noise filtering or edge detection.

The theory behind this transform and its applications has been exposed by *von Neumann* and *Lafe* [1],[2]. According to them, there are a set of parameters involved in obtaining a single CAT basis. These are: the space dimension (one dimensional $D = 1$ or two dimensional $D = 2$), the size of the basis (N), the starting initial state, also called cellular automaton that can be presented as a vector in $D = 1$ or as a $N \times M$ matrix in $D = 2$.

Other parameters are the evolution time ($T > 0$), the number of states each cell can take ($K > 2$), the rule number according to which the initial state will evolve (RN), the number of neighbors considered in the evolution (m), the *Class* (=1 in case $T = N$ and Class = 2 in case $T > N$), and the *Type* of linear development for the obtained evolution in order to calculate the basis. See equation 4 for an example.

All parameters and their values for the experiments in this paper are listed in table I and II. In general, their amount and variability range allows obtaining a huge number of bases and, among these, the one which fits better to the problem in question than when using other transforms.

2 Experiments.

2.1 Obtaining the basis

If f_i is a discrete signal with $i = 1, \dots N$ then a basis $A_{i,k}$ is search so that

$$f_i = \sum_k c_k A_{i,k} \qquad (1)$$

where c_k are the coefficients obtained from the inverse transformation;

$$c_k = \sum_i f_i B_{i,k} \qquad (2)$$

and $B_{i,k}$ is the inverse of $A_{i,k}$.

The construction of a basis starts with the initial state (*IS*) of the automata. In case ($D = 1$) it can be described by an N - element vector $a(i, t)$ with $t = 0$ and $i = 1, \dots, N$, and in $D = 2$, by a $N \times M$ matrix which elements are $a(i, j, t)$ with $t = 0$, $i = 1, \dots, N$ and $j = 1, \dots, M$. They are both composed of cells linked to each other and each one taking a finite number of K possible states (values). Figure 1 shows an initial state with $D = 1$, $N = 4$ and $IS = [1\ 0\ 1\ 1]$.

Fig. 1 Cellular Automaton.

The future state of each cell at $t > 0$ will be determined by an evolution rule and by the present states of the automaton in a certain neighbourhood m. Where a_{it} is the i cell at state t. $a_{it} = a(i, t)$, i = space index, t = time index. The initial state will evolve T times to build a $N \times T$ matrix in 1-D case according to an specific evolution rule described by a rule called *Wolfram* rule. See equation 3. In *Class* =1 evolutions, the evolution time $T = N$ which returns square matrices. With these matrices and using equation 5, the transformation basis can be obtained.

The *Wolfram* vector W_j ($j = 0, 1, 2, \dots, 2^m$), is a weight vector obtained form the rule number (*RN*), the maximum state number of each cell and the number of neighbours to consider in evolution (m). Its precise expression can be found in [1]. Example: $W = [0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1]$, is the *Wolfram* vector for $RN = 53$, $K = 2$ and $m = 3$. The evolution of cell a_{it} at time $t+1$, for 3 neighbours, is described by the following rule:

$$a_{it+1} = (W_0 a_{0t} + W_1 a_{1t} + W_2 a_{2t} + W_3 a_{0t} a_{1t} + W_4 a_{1t} a_{2t} + W_5 a_{2t} a_{0t} + W_6 a_{0t} a_{1t} a_{2t} + W_7)^{[illegible]} \bmod K \qquad (3)$$

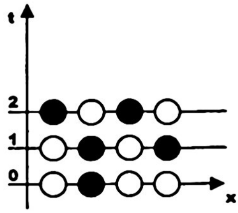

Fig. 2 Evolution of initial state.

The evolution leads to a $N \times T$ matrix E used to form the desired basis according to some lineal combination:

$$A(i,j) = \alpha + \beta \cdot E(i,j) \qquad (4)$$

where α, $\beta \in \Re$ are real number introduced by the user.

In 2-D case, the basis can be obtained as a result of linear evolutions of two dimensional initial states or as an inner product of one dimensional basis already obtained:

$$B(i,j,k,l) = A(i,j) * A(k,l) \qquad (5)$$

All bases used in this work were obtained using the second method. *Gramm-Schmidt* procedure was performed over those bases that result not orthogonal after an orthogonality test [5].

2.2 Ordering the basis

The orthonormal bases were placed ordered in a set according to its performance measured in the entropy values of the coefficients obtained with each of them. According to the relation between *Shannon* entropy [6] and compression ratio of a data set, only when entropy based algorithms are used, bases among the first positions will report the best results in compression purposes. In this particular case bases of dimension 4x4x4x4 ($N = 4$) and dimension 8x8x8x8 ($N = 8$) were studied.

Equation 7 shows an example of a basis of size 2x2x2x2 ($N = 2$),

$$A(i,j,k,l) = \begin{bmatrix} a_{ij11} & a_{ij12} \\ a_{ij21} & a_{ij22} \end{bmatrix}, \quad a_{ij11} = \begin{bmatrix} a_{1111} & a_{1211} \\ a_{2111} & a_{2211} \end{bmatrix}, \quad i,j,k,l = 1,2. \qquad (7)$$

When dealing with cellular automata bases is common to obtain the same basis using different set of parameters which justifies their widespread use in other applications like encryption for example. For compression purposes on the other hand, only different bases are required and among them, the one with better performance. So, in order to have only those different to each other, only the new and different to those already in, are added to the set.

2. 3 Bases classification.

The process is done transforming a set of real images with every basis in the set and obtaining the entropy values of their coefficients transformed. The bases are ordered in the set according to the transformed images coefficients entropy values. So is possible to select among the first ones, the best for compression purposes. In this case a lossless arithmetic compression algorithm has been used.

Table 1. Parameters for obtaining basis $N = 4$ (left) y $N = 8$ (right)

Basis	RN	IS				Basis	RN	IS							
1	28	1	0	1	1		11	0	1	1	1	1	0	1	0
2	10	0	0	0	1	2	15	0	1	1	0	1	1	1	1
3	15	0	0	0	1	3	15	0	1	1	1	1	0	1	1
4	217	0	0	0	1	4	11	0	0	0	0	1	1	0	0
5	78	0	0	0	1	5	15	1	0	0	0	0	1	0	0
6	130	0	0	1	0	6	15	1	0	0	1	0	0	0	0
7	16	0	0	1	0	7	47	1	1	0	0	0	0	0	0
8	167	0	0	1	0	8	43	1	1	1	0	0	1	1	1
9	252	0	0	1	0	9	43	1	1	1	1	0	0	1	1
10	59	0	0	1	0	10	15	1	1	1	1	0	1	1	0
11	173	0	0	1	1	11	11	0	0	0	1	1	0	0	0
12	230	0	1	1	0	12	15	0	0	0	1	1	1	0	1
13	74	0	1	1	0	13	14	0	0	1	0	1	0	1	1
14	188	1	0	0	1	14	15	0	1	0	1	1	1	0	0

Table 2. Other parameters

N	D	IS	RN	Class	T	Type	K	m	α	β
variable	2	variable	variable	1	N	2	2	3	-1.0	2

But first one has to be sure that there is a different performance between them no matter what images they are processing. Ten real images (256 x 256, 8bpp) were transformed with two different sets of bases with $N = 4$ and $N = 8$, and the entropy values of the resulting coefficients were calculated.

Table 3. Bases with $N = 4$ (Left) and $N = 8$ (Right). Entropy and *PSNR* after reducing the amount of necessary bits from 10 to 6 ($N = 4$) and from 14 to 8 ($N = 8$).

Basis (KN)	Entropy [bpp]	PSNR [dB]	Basis (IS)	Entropy [bpp]	PSNR [dB]
28	1.81	46.7	122	4.86	51.1
10	3.89	56.1	111	5.35	51.8
15	4.08	52.3	123	5.23	51.4
217	3.98	56.6	12	1.04	27.3
78	1.93	46.5	132	2.26	35.9
130	3.94	56.1	144	1.45	30.8
16	3.94	56.1	192	1.08	27.6
167	2.96	43.4	231	4.86	51.1
252	4.08	51.9	243	5.12	51.5
59	1.88	46.4	246	5.35	51.8
173	1.81	46.7	24	1.39	29.7
230	1.69	43.7	29	1.40	30.3
74	1.83	46.9	43	5.11	51.5
188	1.83	46.9	92	2.26	35.9

The bases in the paper had been obtained using the parameters in tables I and II. The initial states are one dimensional but using the equation 5 two dimensional bases can be obtained from one-dimensional ones. *Type* = 2 means a linear expression like the one in equation 4, $K = 2$ is the maximum number of states of each cell; also [0, 1].

In figure 3 the lines does not cross each other. This means that there is an independent behaviour or different performance of each basis for any image. The expression used for entropy is *Shannon* entropy:

$$E = -\sum_{i}^{N} p_i \log_2(p_i), \quad i = 1,\ldots, N \quad [bpp]. \qquad (7)$$

where p_i is the probability for one specific value in the coefficients and N is the number of coefficients.

The basis with $N = 8$ showed similar behaviour that those with $N = 4$. It has been shown that their performance is different from each other and independent for each image.

3. Results and discussion.

In order to represent coefficients obtained with bases of $N = 4$, 10 *bpp* (bits per pixel) were needed, (2 *bpp* more than in the original image), that is 131072 bits more then the original image, while in transforming the images with bases of $N = 8$, 14 *bpp* were needed.

Table 4. Entropy and compression ratios for bases with $N = 4$ (left) and $N = 8$ (right)

#	RN	Entropy [bpp]	TC [bpp]	PSNR [dB]	#	RN	Entropy [bpp]	TC [bpp]	PSNR [dB]
8	*167**	*0.81*	*0.79*	*23.2*	*4*	*12**	*0.43*	*0.40*	*21.5*
12	230	1.03	0.92	23.9	*7*	*192**	*0.50*	*0.47*	*21.1*
11	173	1.06	0.80	27.7	*11*	*24**	*0.73*	*0.69*	*22.8*
1	28	1.07	0.80	27.7	*12*	*29**	*0.74*	*0.70*	*22.9*
10	59	1.1	0.88	25.4	*6*	*144**	*0.79*	*0.75*	*22.8*
14	183	1.17	0.93	26.4	*14*	*92**	*1.32*	*1.22*	*27.6*
13	74	1.17	0.94	26.4	*5*	*132**	*1.32*	*1.22*	*27.6*
5	78	1.25	1.03	25.4	8	231	1.94	1.27	31.0
2	10	2.81	2.0	34.8	1	122	1.94	1.29	31.0
7	16	2.81	2.02	35.0	9	243	2.09	1.41	30.7
6	130	2.81	2.06	35.0	13	43	2.09	1.42	30.7
4	217	2.86	1.99	35.7	3	123	2.17	1.47	29.7
3	15	2.92	1.78	29.8	2	111	2.25	1.58	30.1
9	252	2.95	1.87	29.6	10	246	2.25	1.58	30.1

The standard deviation between original and reconstructed images after rounding the coefficients was only 0.28 units in all cases, i.e. to a *PSNR* of 60 units. This means that the simple rounding of the coefficients does not introduce errors in the reconstructed image.

The amount of necessary *bpp* was reduced to 6 in bases with $N = 4$ (except in basis with $RN = 167$ where 10 *bpp* were needed). On the other hand in bases with $N = 8$, this quantity was reduced to 10 and 8 *bpp* in some cases from 14 *bpp* originally necessary to represent the transformed coefficients. These reductions brought considerable entropy decrease of the coefficients. A non-uniform coder was also introduced who reduced more the coefficients entropy values while maintaining the *PSNR* in acceptable values i.e. with very small loss in the reconstructed image. See table III.

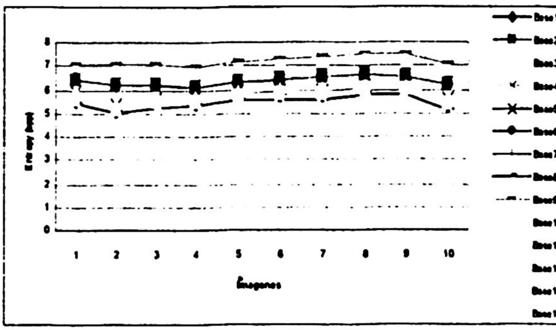

Fig. 3 Cellular automata transform bases performance measured in the entropy values of the coefficients of 10 different images. Each basis shows a stable behaviour in comparison to the others, therefore is possible to achieve better results with a certain basis no matter what the input image is. Bases with N = 8 show similar behaviour.

The ordered basis according to the mean entropy value, the *PSNR* values and the mean compression ratio over the set images are shown in table IV. The minimal loss in the reconstructed images is due to the presence of quantization.

The entropy values were calculated as the mean value for each basis of all image transformed coefficients using expression (7). Compression ratio has been calculated as the quotient of necessary bits to represent the original image and the amount of bits necessary to represent the codified image.

The *Peak Signal-to-Noise Ratio (PSNR)* was calculated using expression

$$PSNR = 10 \cdot \log_{10}\left(\frac{x_{pp}^2}{MSE}\right) \quad [dB] \cdot \qquad (8)$$

where $x_{pp} = 255$,

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - x'_i)^2 \qquad (10)$$

is the Mean Square Error and x' is the reconstructed image.

As explained before, the number of basis that can be obtained varying the parameters is a huge number. In this paper we had limit this number to a few bases in order to prove the relation of entropy and compression ratio when using entropy based coders.

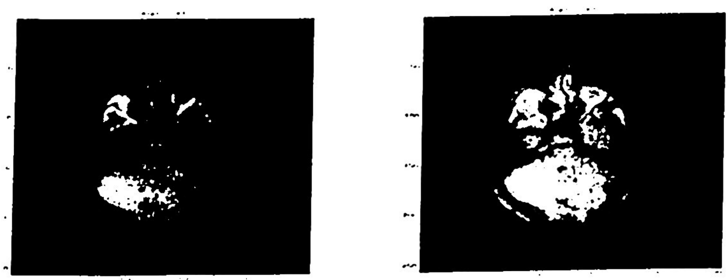

Fig. 4 a) Image a) Magnetic resonance image (T1slice), 256×256 pixels and 8 bpp, b) reconstructed image after being compacted: *PSNR*= 25.4 dB, TC = 0.39 bpp (CR = 20.2 times).

The entropy based coders use the probability of occurrence of a value in the transformed coefficients, while the energy based coders take account of the coefficients magnitude. Therefore when using energy based coders or combining these with entropy based coders, the criterion exposed here will not be effective in selecting the best basis for compression purposes.

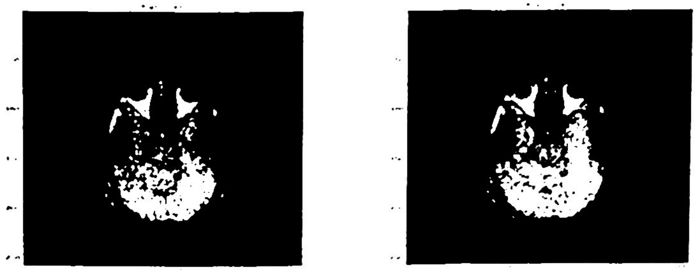

Fig. 5 a) Magnetic resonance image (T1slice). 256×256 pixels and 8 bpp.
b) reconstructed image after being compacted *PSNR*= 27.1 dB,
TC = 0.77 bpp (CR = 10.3

Figures 4 and 5 show two images compressed using base named (*IS* = 12) in table IV. The images are 256x256 pixels and 8 *bpp*. The compression rates obtained are between 10 and 20 times, keeping the *PSNR* bigger then 25 *dB*, what is acceptable for processing medical images.

References

[1] J. von Neumann, Theory of Self-Reproducing Automata (edited and completed by Arthur Burks), University of Illinois Press, 1966.

[2] O. Lafe, Cellular Automata Transforms: Theory and Applications in Multimedia Compression, Encryption, and Modeling, Kluwer Academic Publishers, 2000, ISBN 0-7923-7857-1.

[3] R.R. Coifman, M.V. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection". *IEEE Trans. on Inf. Theory*, Vol.38, No.2, 1992.

[4] I.H. Witten,, R.M. Neal, J.G. Cleary, "Arithmetic coding for data compression", *Communications of the ACM*, Vol.30, No.6, June 1987, pp. 520 - 540.

[5] G. Arfken, "Gram-Schmidt Orthogonalization", in *Mathematical Methods for Physicists*, 3rd ed,. Orlando, FL: Academic Press, pp. 516-520, 1985.

[6] C.E. Shannon, "A mathematical theory of communication",. *Bell System Technical Journal*, Vol. 27, 1948.