Pergamon

PII:S0952-1976(97)00040-7

Engng Applic. Artif. Intell. Vol. 10, No. 6, pp. 581–591, 1997

Printed in Great Britain
0952-1976/97 $17.00+0.00

Contributed Paper

Data Compression and Encryption Using Cellular Automata Transforms

OLU LAFE

Lafe Technologies, Cleveland, OH, U.S.A.

(Received December 1996; in revised form April 1997)

A family of basis functions, generated from the evolving states of Cellular Automata (CA), is used to compress and encrypt data. The operations required in encoding and decoding the data are described under the umbrella Cellular Automata Transforms (CAT). There is a huge number of these transform bases. CAT can be used in the way other mathematical transforms (e.g., Fourier, Discrete Cosine, Laplace, Wavelet, etc.) are utilized. In data-compression applications, the rules and pertinent keys used to generate the CA are selected to favor those that yield basis functions with the best information-packing characteristics. On the other hand, for encryption the selection is biased towards those with the tendency to yield an avalanche effect. In the latter case the transform process must be error-free.

Keywords: Cellular Automata, Cellular Automata Transforms, data compression, encryption.

1. INTRODUCTION

Considerable interest has been shown in the past decade in the use of cellular automata for computing, or as models of physical, chemical, biological or information processes. The main challenge is to associate a given phenomenon with the evolving automata field. Traditional approaches, which include so-called lattice-gas models, (Dab *et al.*, 1991; Delahaye, 1991; Despain *et al.*, 1990; d'Humiéres and Lallemand, 1987; Di Pietro *et al.*, 1994; Frisch *et al.*, 1986; Frisch *et al.*, 1987; Frisch, 1991) demand the use of a large number of computational cells. More complications are introduced by the convoluted manner in which the CA space and its evolution is linked with the particular process.

CAT does not require a huge number of computational cells. There is also a more direct way of achieving the linkage between the CA space and the process. CAT incorporates a huge set of orthogonal, semi-orthogonal, bi-orthogonal and non-orthogonal bases. These can be adapted to the peculiarities of a given problem.

This paper presents a brief overview of CAT and showcase applications in digital image compression and encryption. A more-detailed description of CAT, including the theory and applications in process modeling, solution of differential/integral equations, data compression and encryption, is contained in a forthcoming book by the author (Lafe, 1998). The following section deals with CA basics, and is intended for readers who are not conversant with the fundamentals of cellular automata.

Correspondence should be sent to: Dr Olu Lafe, Lafe Technologies, 25988 Highland Rd., Cleveland, OH 44143, U.S.A. [E-mail: lafe@en.com].

2. CA BASICS

2.1. Definition

Cellular Automata are dynamic systems in which space and time are discrete (Gutowitz, 1991). The cells, which are arranged in the form of a regular lattice structure, have a finite number of states. These states are updated synchronously according to a specified local rule of interaction. For example, a simple 2-state 1-dimensional cellular automaton will consist of a line of cells/sites (Fig. 1). A given node can be either **on** (assigned a state value 1) or **off** (value 0). The closest nodes to node P are those to its immediate left and right. In that case, one can have a local neighborhood of three cells. The state of P at time $t+1$ will be determined by the states of the cells within its neighborhood at time t.

Another structure is that of a square lattice, depicted in Fig. 2. The intersections of the squares form the nodes of the automata. The closest nodes to a given node are the four to

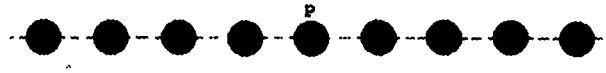

Fig. 1. One-dimensional cellular space.

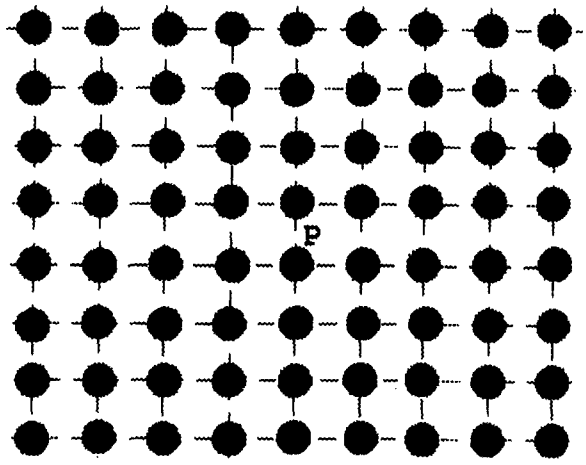

Fig. 2. A two-dimensional square lattice space.

the immediate North, South, West and East, moving along the lines connecting the nodes.

The specified node, P, with its four nearest neighbors, form the *von Neumann* neighborhood. Again, the state of the given node at time $t+1$ will be determined from the states of the nodes within its neighborhood at time t. The rule of evolution is applied to all nodes (and their associated neighborhood) at the same time. The so-called *Moore* neighborhood incorporates the *von Neumann* and the diagonal nodes. The *von Neumann* neighborhood for a cellular space drawn from a diamond lattice is shown in Fig. 3. In this case, a possible neighborhood is that of the node P and the six closest nodes.

Using a specified rule (usually deterministic), the values are updated synchronously in discrete time steps for all cells. With a k-state automaton, each cell can take any of the integer values between 0 and k-1. In general, the rule governing the evolution of the cellular automaton will encompass r sites up to a finite distance away. The cellular automaton is described as a k-state, r-site neighborhood CA.

2.2. Nomenclature

Consider a one-dimensional, dual-state, 3-site neighborhood CA. The state of each cell is given by the Boolean variable **a**. When the state is **on**, $\mathbf{a}\equiv 1$. Otherwise it is **off** and $\mathbf{a}\equiv 0$. The quantity $\mathbf{a}_{it}$ represents the state (Boolean) of the i-th cell, at discrete time t, whose two neighbors are in the following states: $\mathbf{a}_{i-1t}$, $\mathbf{a}_{i+1t}$. In general, a rule is needed, that will be used to synchronously calculate the state $\mathbf{a}_{it+1}$ from the states of the cells in the neighborhood at the t-th time level. The cellular automaton evolution is expressible in the form:

$$a_{it+1}=\mathbf{F}(a_{i-1t},a_{it},a_{i+1t}) \tag{1}$$

where F is a Boolean function defining the rule. There are $2^3=8$ possible configurations for each neighborhood in a dual-state 3-site neighborhood automaton. These are: $111\rightarrow C_0$, $110\rightarrow C_1$, $101\rightarrow C_2$, $100\rightarrow C_3$, $011\rightarrow C_4$, $010\rightarrow C_5$, $001\rightarrow C_6$, $000\rightarrow C_7$ in which C_n is the Boolean value generated by the rule given the n-th configuration. There are 2^8 rules for the 2-state/3-site CA. The convention (Wolfram, 1983) is to assign the integer R to the rule generating the function F such that:

$$R=\sum_{n=0}^{7} C_n 2^n.$$

Hence, R takes on a value between 0 and 255 for a 2-state/3-site CA. In general, for a k-state/r-site CA, there are k^{k^r} rules, and the evolution is expressible in the form.

$$a_{it+1}=\mathbf{F}(a_{i-mt},a_{i-m+1t},\cdots,a_{i+mt}) \tag{2}$$

where $m=(r-1)/2$, assuming r is odd. If there are N cells in the entire one-dimensional space, there is a total of k^N possible initial configurations with which to start the evolution of the CA. Furthermore, if the CA is run over T discrete time steps, the number of boundary (at the left and right) configurations possible is k^{2T}. Since there are k^{k^r} rules, the number of ways a k-state/r-site/N-cells CA can evolve over T time steps is of the order k^{k^r+N+2T}.

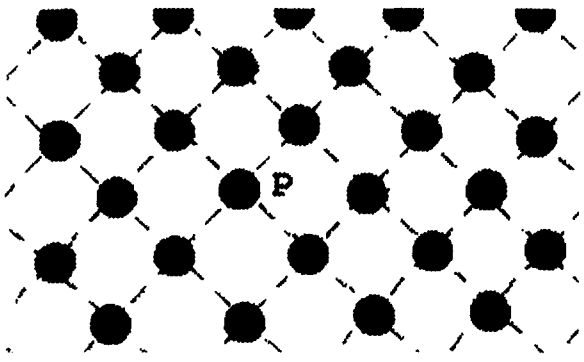

Fig. 3. A two-dimensional diamond cellular space.

3. THEORY OF CAT

Given a process described by a function f, defined in a physical space of a lattice grid i, basis functions A and their associated transform coefficients c are required, defined in the cellular automata frequency space k, which satisfy

$$f_i=\sum_k c_k A_{ik}. \tag{3}$$

The basis functions are related to the evolving field (*i.e.*, the states) of the cellular automata. Note that each point on the physical grid i has an associated basis function A (spanning the entire physical space i and CA space, k). Equation (3) represents a mapping of the process f (in the physical domain) into c (in the cellular automata domain) using the building blocks A as transfer functions. In many applications one needs to obtain transform coefficients c, with properties not necessarily possessed by the original function, f. Alternatively, the transformation process should reveal things about f not readily observed in the physical domain.

For example, in data compression applications, the transformation should reveal any redundancies in the original data. The elements of c with insignificant or zero magnitudes reveal the degree of redundancy detected by the CA transform. Among other things there is a need for:

- basis functions that will maximize the number of negligibly small c coefficients, while minimizing the encoding error. The objective is to find bases that produce coefficients with the minimum entropy. The entropy of the transform coefficients is:

$$E=-\sum_i \phi_i \log_2(\phi_i) \tag{4}$$

  where ϕ_i is the probability that a transform coefficient is of magnitude i.
- a small alphabet base for generating the basis functions, A;
- considerable ease in calculating the transform coefficients, c;

For an encryption application the plaintext is f and the ciphertext is c. The representation in equation (3) should possess the so-called *avalanche effect*: a small change (*e.g.*, 1 bit) in f should result in a big change in c; and a small change in the keys used in generating A should yield a significant change in the ciphertext. In this case, one actually seeks to maximize the alphabet base used in generating the basis functions. This will help thwart the efforts of code breakers. Furthermore, the CA transforms of the data must involve non-floating-point calculations, and no error can be tolerated in the encoding/decoding phases.

The essence of CAT is that one can always find CA rules (and the associated neighborhood, initial/boundary configuration, lattice arrangement, etc.) which will result in basis functions and transform coefficients with the properties desired for a given problem. The chief strength is the huge number and varied nature of the basis functions available.

3.1. CA basis functions

The distinguishing characteristics of CA basis functions include:

(1) The method used in calculating the bases from the evolving states of cellular automata.
(2) The degree of orthogonality (full, semi, bi or none) of the basis functions.
(3) The support of the bases. Windowed basis functions have a compact support (these are zero everywhere except within a defined region of space and frequency).
(4) The method used in calculating the transform coefficients.

The simplest bases are of the β-group. The elements are integers $[-1,1]$ (*non-windowed*) and $[-1,0,1]$ (*windowed*). Next are ρ-bases, whose elements are generated from the instantaneous point density of the evolving field of the cellular automata. The basis functions, called μ-bases, are generated from a multiple-cell-averaged density of the evolving automata. S-bases (which are derived from β, ρ and μ groups) are usually non-orthogonal, and are excellent for self-generating transformations of data.

One-dimensional cellular spaces offer the simplest environment for generating CA transform bases. They offer several advantages, including:

- A manageable alphabet base for small r and small k. This is a strong advantage in data-compression applications.
- The possibility of generating higher-dimensional bases from combinations of the one-dimensional base.
- The excellent knowledge base of one-dimensional cellular automata.

In a 1-D space consisting of N cells, the transform base

$$\mathbf{A}\equiv A_{ik}\ i,k=0,1,\cdots N-1.$$

Consider the data sequence $f_i(i=0, 1, 2,\ldots N-1)$.

$$f_i=\sum_{k=0}^{N-1} c_k A_{ik}\ i=0,1,2,\cdots N-1 \tag{5}$$

in which c_k are the transform coefficients. There are infinite ways by which A_{ik} can be expressed as a function of $\mathbf{a}\equiv a_{it}$, $(i, t=0, 1, 2,\ldots N-1)$. A few of these are enumerated below for a dual-state CA. Note the presence of variable parameters in some of the formulas for these bases. In such cases there are multiple sub-bases, each with unique characteristics and capable of being used solely, or in a group, as transform bases.

- **Type 1**: $A_{ik}=2a_{ik}-1$
  where a_{ik} is the state of the CA at the node i at time $t=k$.
- **Type 2**: $A_{ik}=2a_{ik}-1$
- **Type 3**:

$$A_{ik}=\mu_{ik}\mu_{ki} \tag{6}$$

$$\mu_{ik}=\frac{1}{2n_w}\sum_{l=i-n_w}^{i+n_w} a_{l_w,k}$$

$$l_w=\begin{cases} 1 & if\quad 0\le l\le N-1 \\ l+N & if\quad l<0 \\ l-N & if\quad l>N-1 \end{cases}$$

  where $1\le n_w\le N-2$.
- **Type 4**:

$$A_{ik}=\rho_{ik}\rho_{ki} \tag{7}$$

$$\rho_{ik}=2a_{ik}-1+\rho_{ik-1}$$

$$\rho_{i0}=2a_{ik}-1$$

- **Type 5**:

$$A_{ik}=\rho_{ik}\rho_{ki} \tag{8}$$

$$\rho_{ik}=2a_{ik}-1+\rho_{ik-1} \bmod L_w$$

$$\rho_{i0}=2a_{ik}-1 \bmod L_w$$

in which $L_w>=2$ is an integer. There are as many ways of generating Type 5 bases as there are for selecting L_w.

- **Type 6**:

$$A_{ik}=\mu_{ik}\mu_{ki} \tag{9}$$

$$\mu_{ik}=\mu_{ik-1}+a_{ik}/\sigma_k$$

$$\mu_{i0}=a_{ik}/\sigma_k$$

$$\sigma_k=1+\sum_{i=0}^{N-1} a_{ik}$$

- **Type 7**: On account of their orthogonality for a large class of CA rules and initial/boundary configurations, the most robust CA bases are those defined with a window:

$$A_{ik}=\delta_u A^*_{uk} \tag{10}$$

$$u=u(i,k)$$

$$\delta_u=\begin{cases}1 & if\ 0\le u<N_w\\ 0 & otherwise\end{cases}$$

where A^* is akin to the bases obtained using the non-windowed types. For example, if A^* is derived from Type 3

$$A^*_{uk}=\mu_{uk}\mu_{ku} \tag{11}$$

$$\mu_{uk}=\frac{1}{2n_w}\sum_{l=u-n_w}^{u+n_w} a_{uk}.$$

3.1.1. Standard orthogonal CA bases

If the bases A_{ik} are orthogonal, then:

$$\sum_{i=0}^{N-1} A_{ik}A_{il}=\begin{cases}\lambda_k & k=l\\ 0 & k\neq l\end{cases} \tag{12}$$

where λ_k $(k=0,1,\cdots N-1)$ are coefficients. The transform coefficients are easily computed as:

$$c_k=\frac{1}{\lambda_k}\sum_{i=0}^{N-1} f_i A_{ki}. \tag{13}$$

3.1.2. Progressive orthogonal one-dimensional CA bases

These allow the transform coefficients to be calculated in the progressive manner:

$$c_k=\sum_{i=0}^{N-1} \epsilon_{ik-1}B_{ik}\ k=1,2,3,\cdots N-1 \tag{14}$$

$$c_0=\sum_{i=0}^{N-1} f_i B_{i0}$$

$$\epsilon_{i0}=f_i-c_0A_{i0}\ i=0,1,2,\cdots N-1$$

$$B_{ik}=\frac{A_{ik-1}A_{ki}}{\lambda_k}$$

$$\lambda_k=\sum_{i=0}^{N-1} A_{ik}A_{ki}\ k=1,2,3,\cdots N-1.$$

3.1.3. Two-dimensional bases

In a 2-D square space consisting of $N\times N$ cells, the transform bases $\mathbf{A}\equiv A_{ijkl}$, $(i,j,k,l=0,1,\cdots N-1)$. For the data sequence f_{ij} $(i,j=0,1,2,\cdots N-1)$ one can write:

$$f_{ij}=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} c_{kl}A_{ijkl}\ i,j=0,1,2,\cdots N-1 \tag{15}$$

in which c_{kl} are the transform coefficients. There are two broad approaches for generating two-dimensional CA transform bases:

(1) Using the evolving states derived from two-dimensional cellular spaces. Here, A_{ijkl} are calculated from $\mathbf{a}\equiv a_{ijt}$, $(i, j, t=0,1,2,\cdots N-1)$.

(2) Calculating the canonical products of one-dimensional bases so that $A_{ijkl}=A_{ik}\,A_{jl}$. There are as many canonical 2D bases as there are permutations of 1D basis functions. Furthermore, there is a special **Type 8** two-dimensional basis function which is derived from one-dimensional cellular automata as:

$$A_{ijkl}=\left[\frac{2a_{ik}a_{ki}}{(L_W-1)^2}-1\right]\left[\frac{2a_{jl}a_{lj}}{(L_W-1)^2}-1\right]$$

where $L_{w\ge 2}$ is the number of states of the automaton.

3.2. Examples of orthogonal CA basis

CA keys for generating one-dimensional orthogonal Type 1 β-class non-windowed CA basis functions are shown in Table 1 for $N=8,16$. Cyclic boundary conditions were imposed on a dual-state/3-site CA to generate the states of the automata. The graphical displays of two of these functions are shown in Figs 4 and 5.

An example of the information-packing characteristic of a typical CAT is shown in Fig. 6. The test data, drawn from a digitized grayscale image of a face, is shown in Fig. 6a.

Table 1. Gateway values for other β-transforms

Rule number	Initial configuration	N
14	11010100	8
15	10010000	8
43	01111110	8
47	11110011	8
142	01101010	8
143	01011101	8
158	11011000	8
159	01011101	8
15	0110110001011111	16
142	1101011010100101	16

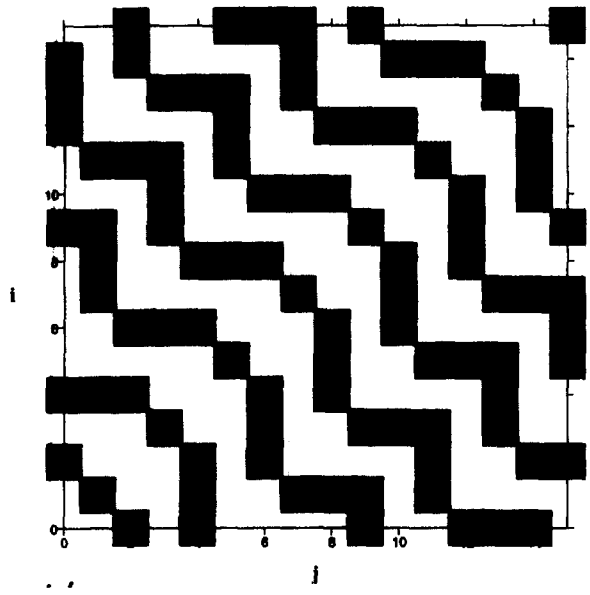

Fig. 4. One-dimensional β-basis function Rule=15; N=16; Init. Config=0110110001011111Bdry. Config=Cyclic; CAT Type=1.

The surface plot of the CA transform coefficients is in Fig. 6b. The standard deviation of the transform coefficients, obtained from a large library of subimages, is shown in Fig. 6c. Note the concentration of energy on a few coefficients.

4. APPLICATIONS

4.1. Image compression

The existence of trillions of transform bases and CA gateway keys mandates different strategies for different data-encoding tasks. For each task, a decision must be made as to how many different CA bases will be used. An evaluation must also be made of the cost associated with each decision in terms of computational cost, encoding/decoding time, and fidelity.

4.1.1. Encoding scheme

A Type 8 β-class two-dimensional CAT was selected to showcase the main features of a CA-based compression. The characteristics of the basis functions are summarized in Table 2.

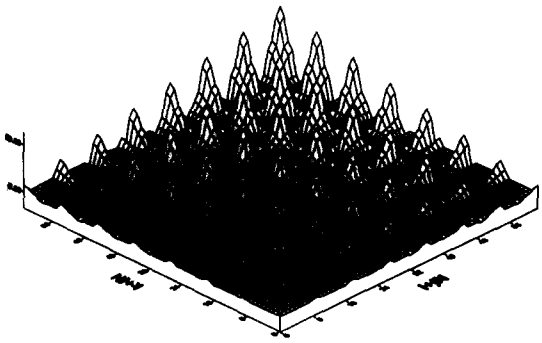

Fig. 5. Two-dimensional ρ-basis functions.

A graphical display of the $64 \times 8 \times 8$ basis functions is shown in Fig. 7. Note how simple these coefficients are (-1s and 1s). The transform process consists of additions and subtractions. The only division involved is when the result of the additions and subtractions is divided by 8, and this is accomplished via bit shifts. The forward and backward transforms are symmetric.

The CAT encoding scheme is hierarchical. Let $w=2^n$ be the width (the number of pixels) of the image while $h=2^m$ is the height, where m,n are integers. Dimensions which are not integral powers of two are handled by the usual zero-padding method. The original image is divided into $2^{(m+n)}/64$ sub-blocks consisting of $8 \times 8 = 64$ pixels. Each block is transformed using the Type 8 β-class basis functions shown in Fig. 7. The transform coefficients c_{kl} fall into four distinct classes (see Fig. 8a): Those at even k and l locations (Group I) represent the low-frequency components. These are sorted to form a new image of size

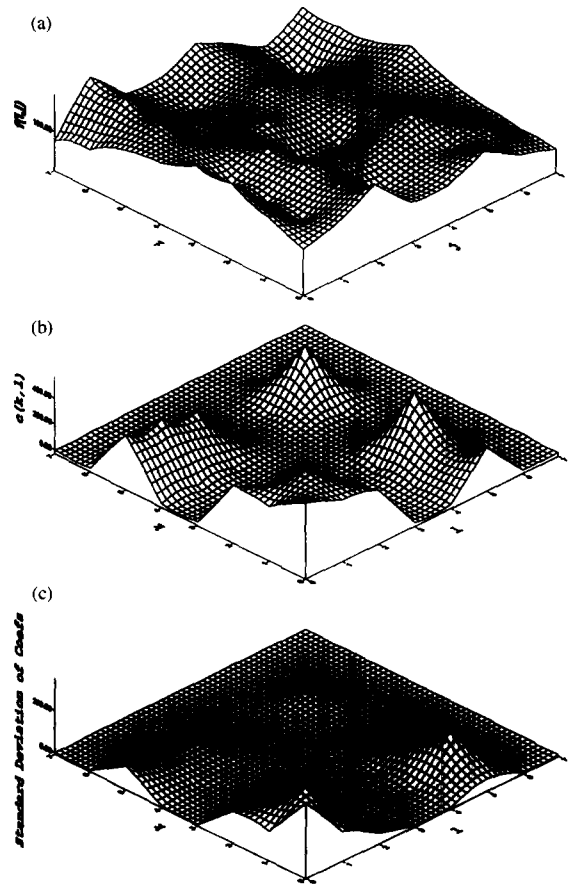

Fig. 6. (a) Test data from a digitized 8-bit image. (b) Surface plot of CAT transform coefficients.(c) Standard deviation of transform coefficients.

Table 2. Summary of CAT used for compression tests

CA basis type	8 *β-class*
Block size	8
Rule number	14
Initial configuration	11010100
Boundary configuration	*Cyclic*

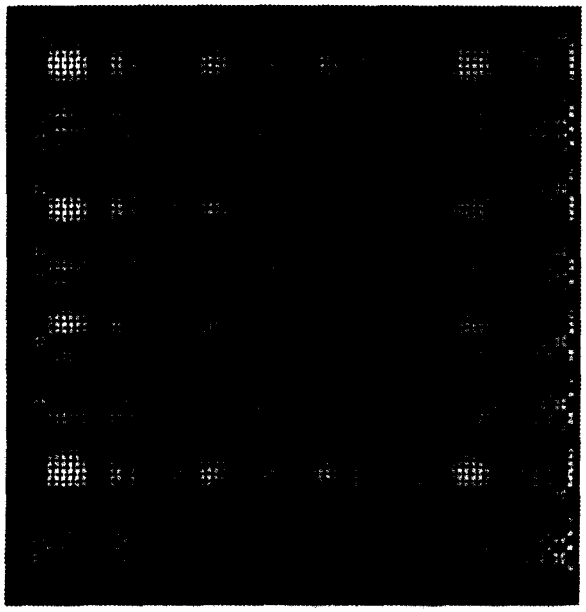

Fig. 7. Type 8 β-class two-dimensional A_{ijkl} basis functions. A_{00kl} is the block at the extreme upper left corner. The top row represents $0 \leq j < 8$; $i = 0$. The left column is $j = 0$; $0 \leq i < 8$. Aij00 is the upper left corner of each block. The white rectangular dots represent 1 (addition) while the black dots are -1 (subtraction).

$2^{(n-1)}2^{(m-1)}$ (at a lower resolution). The rest (Group II: k even, l odd; Group III: k odd, l even; Group IV k odd, l odd) of the coefficients are high-frequency components. In CAT processing these can be shown (see Figs 9a and b) to provide pertinent information on the edges of the image. The low-frequency components can be further transformed by subdividing them into $2^{(m+n-2)}/64$ sub-blocks. The ensuing transform coefficients are again subdivided into 4 groups (Fig. 8b). Those in Groups II, III, and IV are stored, while Group I is further CAT-decomposed and sorted into another 4 groups. For an image whose size is an integral power of 2, the hierarchical transformation can continue until Group I contains only 16 elements (*i.e.* $8 \times 8/4$).

For each block:

$$c_{kl} = \left(\sum_{i=0}^{7} \sum_{j=0}^{7} f_{ij} A_{ijkl} \right) / 8$$

$$f_{ij} = \left(\sum_{i=0}^{7} \sum_{j=0}^{7} c_{kl} A_{ijkl} \right) / 8$$

Compression is greatly enhanced by sorting and storing the transform coefficients, for all resolutions, together within their respective groups. Hence Group II coefficients at the highest resolution (s) are stored next to those from the same group at lower resolution ($s-1$). A fast implementation of CAT permits the transformation to be carried out in O ($Constant \times N_T$) where $N_T = wh$ is the total number of pixels.

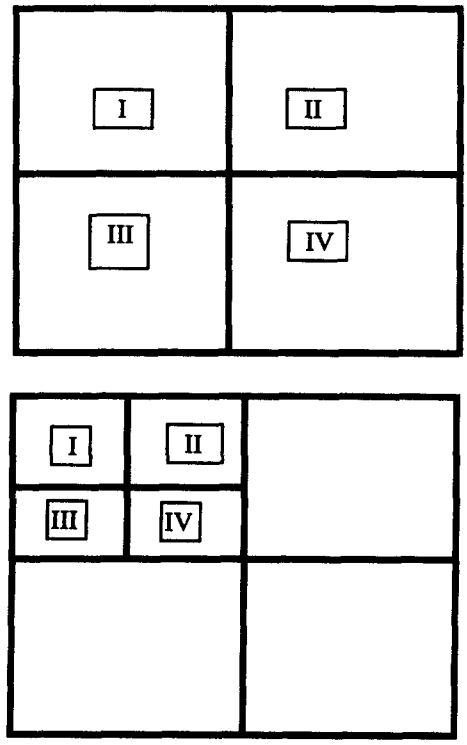

Fig. 8. (a) Decomposition of CAT coefficients into 4 bands. Group I is equivalent to the 'low-frequency'; components. Group II, III, and IV are the 'high-frequency'; components. (b) The image formed by the Group I components in Fig. 8a is further CAT decomposed and sorted into another 4 groups at the lower resolution.

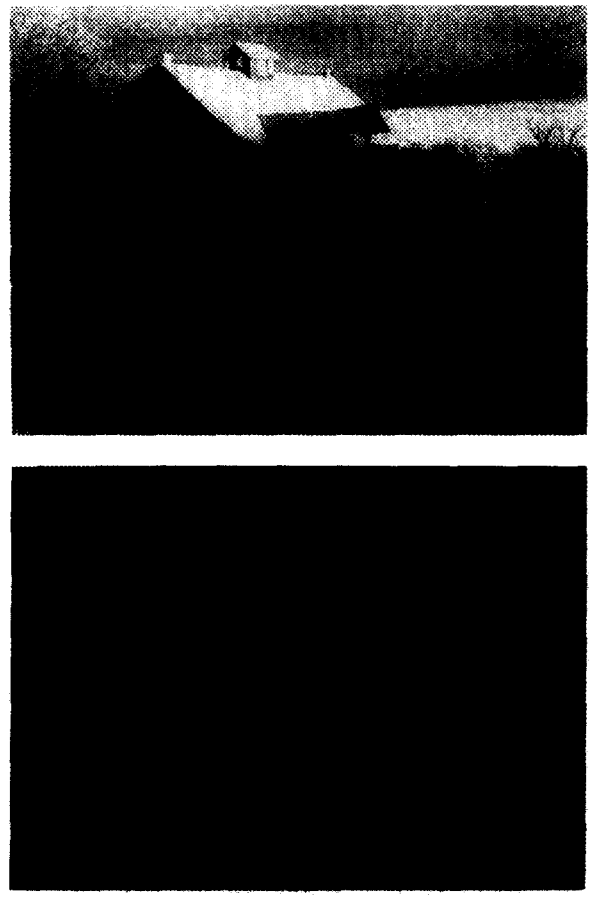

Fig. 9. (a) Original image of a barn. (b) Image reproduced by using only CAT coefficients in Group II, III, IV at the highest resolution.

4.1.2. Quantization scheme

A combination of set partitioning, thresholding, and scalar quantization (the *STQ* scheme) has been found to be most effective (speed and compression efficiency) for this class of hierarchical CAT encoding. The steps involved in the scheme are as follows:

(1) *Partition transform coefficients into* 4 *sets. The first set* $S1$ *consists of the low-frequency (Group I) coefficients at the coarsest level. If the image size is square and the width is an integral power of two,* $S1$ *will consist of* 16 *coefficients. The second set* $S2$ *is made up of Group II coefficients. The ordering is from the coarsest level to the finest resolution. Similarly, set* $S3$ *is derived from Group III coefficients, while set* $S4$ *is from Group IV coefficients.*

(2) *Each of the sets* $S2$, $S3$ *and* $S4$ *is further partitioned into the respective resolution. For example, set* $S2$ *will consist of* $S20$ *(*16 *coefficients) at the coarsest level,* $S21$ *(*64 *coefficients) at the next level, then* $S23$ *(*256 *coefficients),...* $S2r$ *(*4^{r+1} *coefficients).*

(3) *Determine the maximum coefficients* c_{max} *(*S_{gr}*) for each set* S_{gr} *(*$g=1,2,3$; $r=0,1,2,\ldots$ *s). Determine the supreme value* C_{max} *of all* c_{max} *(*S_{gr}*).*

(4) *Select a threshold factor* τ *(*$0 \leq \tau < 1$*) and normalizing coefficient* λ.

(5) *Encode all* 16 *coefficients belonging to set* $S1$ *by dividing each of the coefficients by the normalizing coefficient* λ.

(6) *For each of the subsets* S_{gr} *check if the maximum coefficient is greater than* τC_{max}. *If true encode a YES symbol, and continue to store all the coefficients (divided by the normalizing constant* λ*) in this subset. Otherwise encode a NO symbol, and continue to the next subset.*

(7) *Entropy encode the ensuing quantized coefficients and YES/NO symbols. Arithmetic coding was most favorable for this class of pyramidal CAT processing.*

Obviously compression is increased as λ, and/or τ, are increased. A more complicated implementation of this scheme could make $\lambda = \lambda(S_{gr})$. Alternatively, an embedded zero-tree encoding (see *e.g.* Said and Pearlman (1996); Shapiro (1993)) can be utilized. The *STQ* scheme combines both speed and compression efficiency. Tests show a 10-fold speed advantage over Shapiro's zero-tree scheme for the same compression ratio and a comparable error rate.

4.1.3. Compression results

The charts below Figs 10a and b show the distribution of the transform coefficients at the coarsest level where the low-frequency image consists of 64 coefficients. The 256×256 color Lena image (Fig. 11) was selected for benchmarking the above Type 8 β-class CAT.

The same Lena image was compressed using the JPEG built into the *QuikCAT*® software package from LAFE TECHNOLOGIES. Figures 11–15 depict the comparative performance of CAT and JPEG for a range of compression

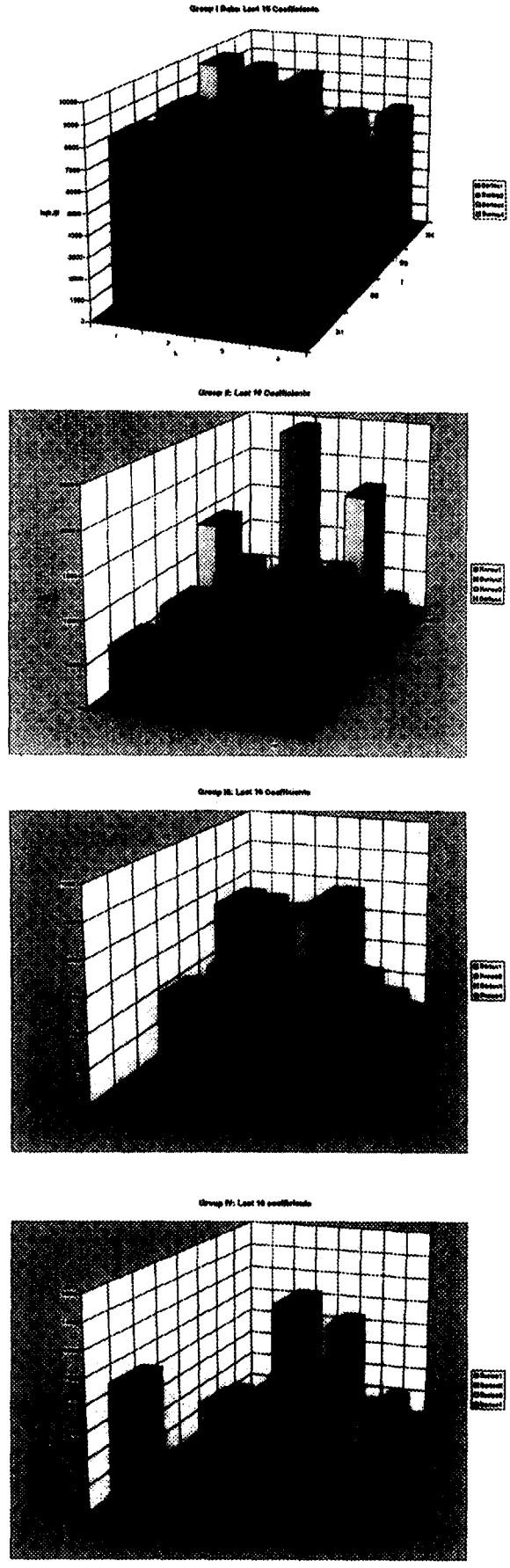

Fig. 10. (a) Group I Data: Last 16 Coefficients. (b) Group II: Last 16 Coefficients.

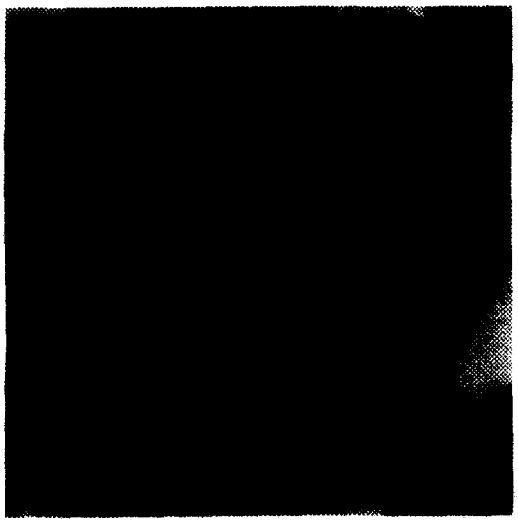

Fig. 11. Original 256 × 256 Lena.

ratios. At low compression ratios (below 20:1) the performance of both schemes is compatible, with a slight edge to JPEG. However, at high compression ratios (>80) the performance of CAT far surpasses JPEG. For example, see Fig. 15a showing a CAT compressed ratio of 130:1, and Fig. 15b with a maximal JPEG compression of 96:1.

5. DATA ENCRYPTION

Given a CAT that is carried out with an N-cell dual-state r-neighborhood one-dimensional CA, a code breaker must contend with searching through:

- 2^{2r} rules;
- 2^{N} initial configurations;
- 2^{2N} boundary configurations;
- N_w window sizes;
- the different types of CA bases.

The odds against code breakage increase tremendously as the number of states, cellular space, neighborhood, and dimensionality increase.

In CAT-based encryption the encoding error must be zero. Windowed progressive bases are the most versatile for

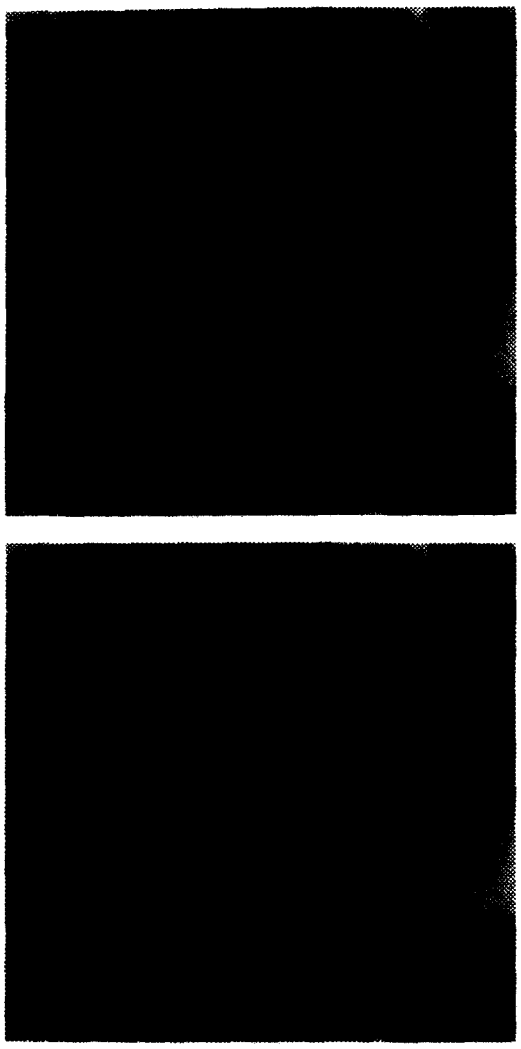

Fig. 12. (a) CAT-Compressed Lena (Cr=25:1). (b) JPEG-Compressed Lena (Cr=24:1).

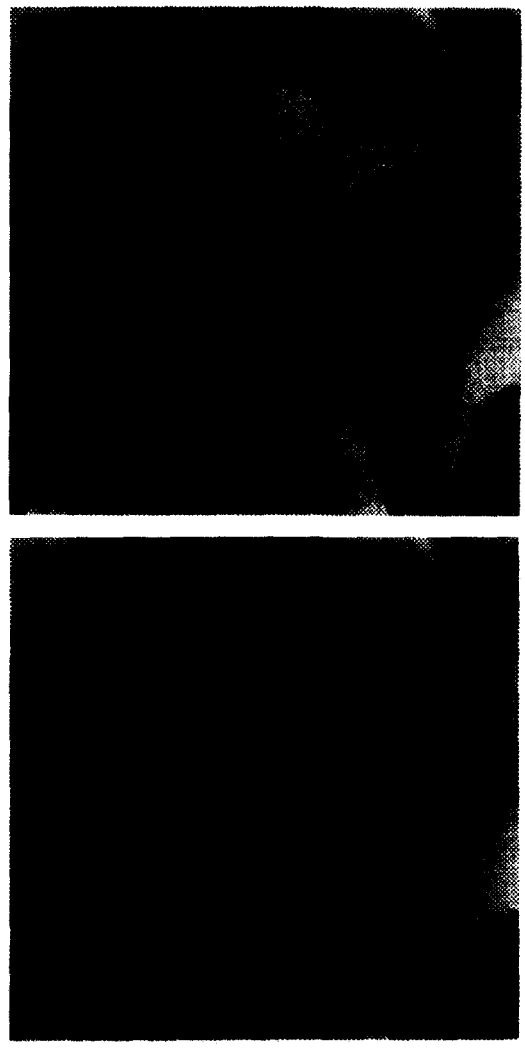

Fig. 13. (a) CAT-Compressed Lena (Cr=51:1).(b) JPEG-Compressed Lena (Cr=51:1).

lossless encoding. The CAT-based encryption (CATE) algorithm, showcased below, has the following features:

- **Symmetric**: The encryption and decryption keys are identical. It is also a **secret-key** algorithm because the sender of the message and the receiver must have sent the key over a secure (preferably different) channel prior to the commencement of the encryption/decryption processes.
- **Block-Based**: The plaintext (original message) and ciphertext (encrypted message) are divided into square blocks of width N—the length of the cellular space. The implementation here uses square blocks of size $N \times N$, although the transform bases are generated from one-dimensional automata. The block width, N, is included in the key.
- **Variable Key Length**: The key length is easily changed through an *embedded* key-generation scheme. Certain parts (not all) of the keys (*e.g.*, N, initial configuration) can be chosen arbitrarily by the user. However, because the transform bases must form an orthogonal set, the program has internal mechanisms to select the remainder of the keys.

5.1. Example

The following message was encrypted using a progressive CAT with $N=16$. The description of the keys is part of the plaintext.

5.1.1. The plaintext

This is the day
The Lord has made
We will rejoice
And be glad in it

This message is being
Encrypted using Cellular Automata Transforms (CAT)
With the following keys:
N=16; Basis Type=6;
Progressive Transform;
Window (Size=11);
Initial Configuration: 1001010011111001

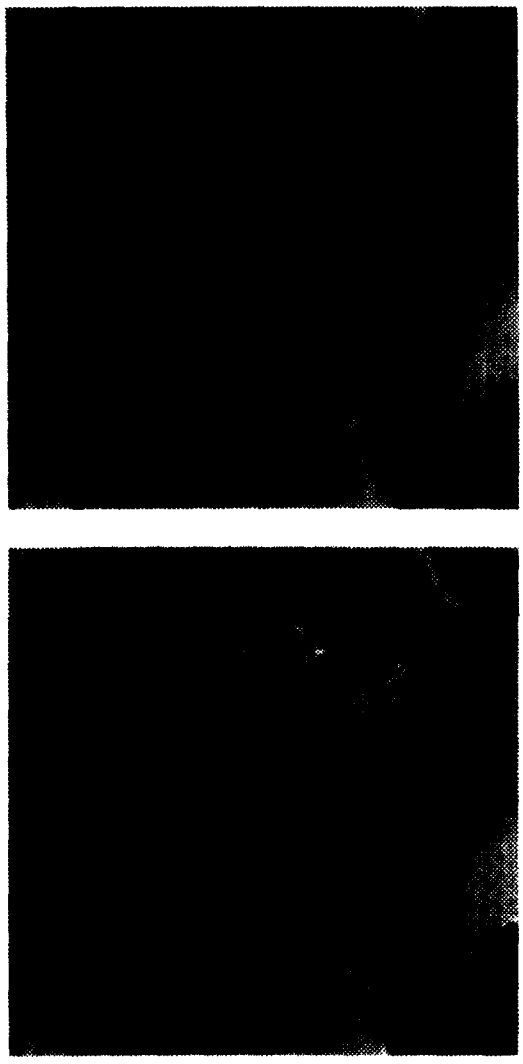

Fig. 14. (a) CAT-Compressed Lena (Cr=80:1).(b) JPEG-Compressed Lena (Cr=79:1).

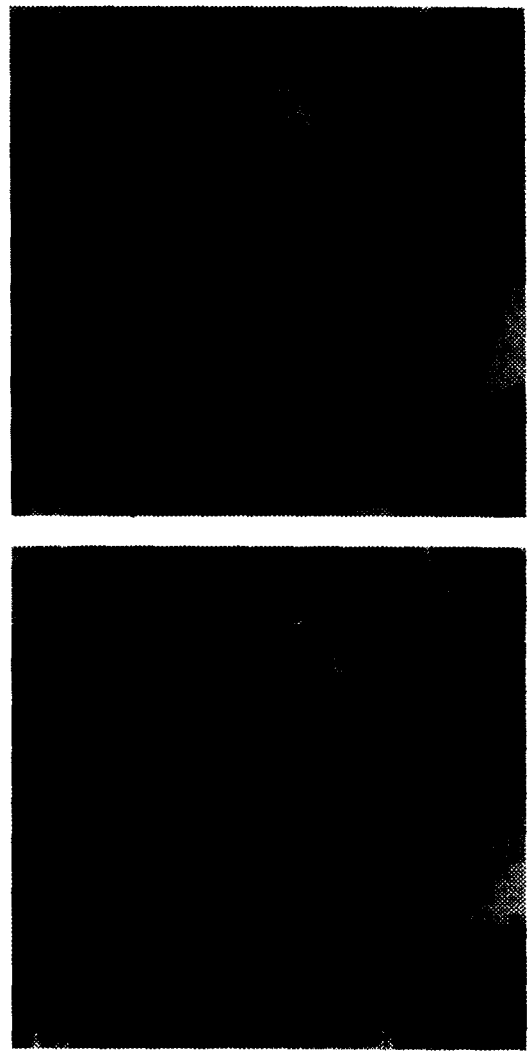

Fig. 15. (a) CAT-Compressed Lena (Cr=130:1).(b) JPEG-Compressed Lena (Cr=96:1).

Boundary Configuration:Cyclic

The actual key file looks like this:
16 11 8 6 91 2 1 1
1001010011111001
1000001101100001
1001010000010000

5.1.2. *The ciphertext**

841041051153210511532A264124671A438A2180169771O1A881084104101327611111
410013105A599A32249A840A955062951011087101321191051O8A29153678A37A255
73029728A4996101106511010032981O1A367124672A41A25632239728A49853210511
61045454545A25961915A241A3743205513994A1244245454545451084104A27513512
92A1387A4895636515164A2766610310132105A5016783A184131216A1801888191A3
538A3288124455114121112116156553721281A825A128633873677A12744A50324668
6A2729889A302A108A612489236A3258A86516079A2929A6441532058247060A2041
72A216A2376699A695A722910213A354423216922A7544A6440743865232969A2408
61920A718A5806917512047A2172A362131841001A6369620791248508A1530145097
776A6254742556114316A471634904A202510876281A6125A266443220986907A1294
186946028A6670879A383424642372A2358A4396A88199934313A56020943268981A1
0759610746725A13587263148885040872605 4A2412A30621A37607061999A63931976
26869A18016A888786027544A3480733163894 40A1564098715922A8296A101138502

2223937236A4450330752362375A24531109202963368734437062019005161500191632
2816228A6098120963976A58164911A165791026921824A98175A85586172631 2A944
4254187021A8369863A17092358283853500000000020A9241A20A1741480000000010
A3111A7A36410000000040A18384A34A3692740000000020A9143A14A19512600000
000A1030A11043A800000000A30130A6111282A136000010A20130A447163A61A609
4260000000071A27A51233A5741562121A2050A35085A12A22455527A204111022A5A5
310A38A5184A103A753854A11A4A8A31A501345A3312246A1216482A1382A202
0A2A3A11A11A93022A81A443770A1570658393564572037112440A76A102142249A
2098A163A3931228A533A745A77A6A74025A104A5732991A10781268006403A270A
278A100A165A36A221A253647636A1999A1512A1556A575A2200A11978593091315
8441A79A8177A11A5562311485223136966616516767111110102A5A476744A243A1
44795A39A299091811859A7492484948491581 98A216A364704750A3245A160814152
785A6021919520661111171 10A10A534928A103138167A8562062281A3045A7063114
971161051A40A7A26A14170A296A9212431662A10378A217045454545A109A165320
168A1054A78240981300A1702570370807A28603454545451A15430890A2210A30429
5747192A43917A10197193637A3591945454545298390A635A65622292267A9390A49
29410153933A1754105061897108321073 10406A1298A183854765633A22984A123271
00788936245160 85A1109284A83A941389A1223A165931303257A11469A1100747758
23176A206765A13973A15786956389012 7A261A287A2218A1115A150536875148A15
389A1566465040342604685790245995517433837111 9842289A126A215A1764850607
0A13363A137204962246949A205212A97931A15538526A3491234A522071186484065
388359A31037026482871A9324A1209639017355 93A160276A75847316891238660832
79825745070247A1602165100524 5A20752A2523059111632 56A2240694758461A1549
96222918705A55357219A2499070060033291A37824027A902A745656A625A2861A3
2271432324566A673052420979A32652973972291A18015490A7606730A1346273A14
7097492405A4895A41589204108160A260373466071 9A1566703817163170A46546314
2073647264644874A1581586210025191923852228323159259201 9A23696A29424485
57A1215755402386 1355130859573187503 41A3650715024187025205754 5A2256318
8000011A2A296A39A11171A13A73932900000011A4A3218A106A1510A185000022A
4A41711A72A18307A33A1289620000011A2A285A32A7129A18A512267000000A2A
1105A51A13255A18A1224516000 0A2A233A8A419A4A1744A189A126000011 63A31
A1717052A890264410A169600003310A19A1414061A847A44A1140960111000A38A2
7195119A1040A3305328A1444705A30341110A5A5A15A1311281A688A269A1940A8
53A476628A5A4A41A16A15223172A1133A70759211864A475648A553A314A3A3A2
0784947A345A2482037767A1990A14A441387272019A6117107A1173A9375955A200
2A1462A640A1528A1405A2833A8345621A36A388529A121A899A387A338A785A80
4A233293A139A100A8833A699A6375867A896941191154A929351129142A44928102
2A16383368A2450592A1833522333261455A921040303

* A is equivalent to the minus sign

The above sample encryption was done with Type 6 windowed (μ-type) basis functions. The basis coefficients are all integers. The length (in bits) of the encryption key in the CATE-algorithm is $L_k = 16 + 6 \log_2 N$. It is based on a dual-state one-dimensional CA. Obviously, as N is increased, the number of states of the automata, the number of cells per neighborhood, the dimensionality of the CA, and so also will L_k increase dramatically. Note that the particular CA keys selected actually expanded the plaintext. The keys can always be selected so that compression is also accomplished along with encryption.

6. CONCLUSIONS

The application of CAT to digital image compression and data encryption is the focus of this paper. The image-compression tests utilize simple dual-state basis functions. A hierarchical, sub-band encoding scheme was used. The ensuing transformation combines the best of desirable advantages such as a simple computation (additions and subtractions as in Hadamard/Walsh transforms) with a sub-band (multi-resolution) filtering (as in Wavelet-based methods). Other applications of CAT include pattern recognition, forecasting, and the solution of partial differential and integral equations. In solving partial differential equations, basis functions with nice differentiation properties are sought. The calculation of the transform coefficients should also not require the explicit inversion of matrix equations. The last feature is provided for by using orthogonal or semi-orthogonal CA basis functions. In dealing with integral equations, a transformation is required, that will result in sparse and diagonally strong coefficient matrices.

REFERENCES

Dab, D., Boon, J.-P. and Li, Y.-X. (1991) Lattice-gas automata for coupled reaction-diffusion equations. *Phys. Rev. Lett.*, **66**(19), 000

Delahaye, J.-P. (1991) *Les Automates*, Pur La Science, pp. 126–134.

Despain, A., Max, C. E., Doolen, G. and Hasslacher, B. (1990) Prospects for a Lattice-Gas Computer. In *Lattice Gas Methods for Partial Differential Equations*, ed. Doolen *et al.*, pp. 211–218. Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.

d'Humiéres, D. and Lallemand, P. (1987) Numerical solutions of hydrodynamics with lattice gas automata in two dimensions. *Complex Systems*, **1**, 599–632.

Di Pietro, L. B., Melayah, A. and Zaleski, S. (1994) Modeling water infiltration in unsaturated porous media by interacting lattice gas-cellular automata. *Water Resource Res.*, **30**(10), 000

Frisch, U., Haaslacher, B. and Pomeau, Y. (1986) Lattice-gas automata for the Navier-Stokes Equation. *Phys. Rev. Lett.*, **56**(14), 000

Frisch, U., d'Humiéres, D., Hasslacher, B., Lallemand, P., Pomeau, Y. and Rivet, J.-P. (1987) Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, **1**, 649–707.

Frisch, U. (1991) Relation between the Lattice Boltzmann Equation and the Navier-Stokes Equations. *Physica D*, **47**, 231–232.

Gutowitz, H. A. (1991) *Cellular Automata: Theory and Experiment*, ed. H. Gutowitz. MIT/North-Holland.

Lafe, O. (1998) *Cellular Automata Transforms: Theory and Applications*. Springer-Verlag. Under Review.

Said, A. and Pearlman, W. A. (1996) A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, **6**(3), 243–250. June

Shapiro, J. M. (1993) Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. On Signal Processing*, **41**(12), 3445–3463. Special Issue on Wavelets and Signal Processing, December

Wolfram, S. (1983) Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, **55**(3), 000