South African Computer Journal
Number 21
August 1998

Suid-Afrikaanse Rekenaartydskrif
Nommer 21
Augustus 1998

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarvereniging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

World-Wide Web: http://www.cs.up.ac.za/sacj/

Subscriptions

	Annual	Single copy
Southern Africa	R50.00	R25.00
Elsewhere	US$30.00	US$15.00

An additional US$15 per year is charged for airmail outside Southern Africa

to be sent to:

Computer Society of South Africa
Box 1714, Halfway House, 1685
Phone: +27 (11) 315-1319 Fax: +27 (11) 315-2276

Cultivating the information systems discipline

Niek du Plooy, Sub-Editor: Information Systems

Whether by 'information system' we mean a simple bookkeeping system for a small business, or a monolithic integrated 'management information system' for a global corporation, all organisations currently need information systems in order to function effectively. The computer and business community at large have readily adopted and accepted the use of the term 'information systems', but perhaps without too much real thought being given to any more profound meaning of the term. Departments bearing that name (or something very similar) are commonly found in organisations. But can the same be said for the 'academic' use of the term, as in describing the information systems discipline? Has it been 'accepted' as a separate scientific discipline?

The term 'discipline' is often loosely applied to indicate the scientific 'field', that is, the organised 'body of knowledge' or 'domains of discourse' within which (mainly) academic activities concerning a specific topic or a number of related topics, are conducted. [3] point out that a scientific discipline has a certain paradigm associated with it, meaning that researchers in that discipline are familiar with the research topics, the research methods and the accepted ways to interpret the results in their chosen field. A discipline is further strengthened and consolidated by the educational process whereby a researcher becomes a practitioner in that discipline, initially through the pursuit of academic degrees and thereafter, through recognition amongst his/her peers. Formal study in a particular discipline results in the value sets and exemplars (the 'paradigm') of that discipline being adopted by the student, either consciously or unconsciously.

Is 'information systems' truly a recognised scientific discipline such such as this? In the past, prominent authors such as Peter Keen did not think so [15, 16]. He deplored the lack of a cumulative tradition and advocated that one be built up, asking for a clarification of the reference disciplines of this new science and a definition of its dependent variable and the building of a cumulative tradition, amongst other things. [1] however, disputed Keen's position and pointed to strong links between research and practice found in their analysis. [11] showed clearly that 'orthodoxy' exists in many aspects of information systems, i.e. in information systems methodologies as well as in other areas of information systems development. This claim was supported by [13] who, in a detailed study based on papers in scientific journals, scientific conferences and textbooks, identified seven different but complementary 'schools of thought' within the field of information systems. In a study of leading universities and leading researchers in decision support systems, [9] provide exemplars, at least for that particular sub-discipline. [5] conclude from a citation study of journal influence during the period 1981 through 1985, that the discipline of information systems has attained stability and that it is in no danger of dying. It seems therefore, that Keen's despair is unfounded and that information systems have indeed grown into a separate, identifiable discipline, even if the field is best described as a 'fragmented adhocracy' ([3]).

The existence of an established scientific community in information systems has been given formal recognition by the recent formation of the Association for Information Systems, a professional society in the tradition of scientific societies, with 1400 members in 35 countries. A recently compiled directory of information systems academics contains entries on some 4,500 researchers from more than 1,000 institutions. A number of basic University and other curricula for information systems education have been published over the years [2, 6, 18]. The most recent of these is Curriculum '95, a joint effort by the ACM, AIS, DPMA, IAIM and ICIS [10, 7]. The most popular discussion group on the Internet (ISWorldNet) devoted exclusively to information systems matters has a membership which in 1997 approached 1829 from 53 countries [14]. A well-defined scientific community therefore exists.

In addition, if the existence of sound academic scholarship is further testimony to the existence of a 'discipline', then information systems can proudly point towards a dramatic growth over the past three decades in the number of scientific journals reporting on research in this area [12]. An even more recent study on research outlets showed that, amongst twenty-seven established journals carrying articles in this field, at least three of the most highly rated top ten are devoted exclusively to the discipline.

Yet, can it be said that the information systems discipline has been conclusively defined and that the research problems and research methodologies prescribed for it have been accepted by all who consider themselves to be working in this field? A re-examination and extension of an earlier (1988) list of keywords for use in classifying information systems literature [4] includes a list of the reference disciplines of information systems, as well as lists of the external environment, the technology, the organisational environment, etc., of information systems. We could argue that this very comprehensive list of keywords (nearly 1300) and other classifications define and describe the discipline of information systems accurately and usefully. For instance, the reference disciplines were listed as:

behavioural science, computer science, decision theory, information theory, organisation theory, management theory, language theories, systems theory, research, social science, management science, artificial intelligence, economic theory, ergonomics, political science, psychology. This list reflects the interdisciplinary or pluralistic nature of information systems.

In the same vein, [19] did a study on the themes of submissions to the journal Information Systems Research and produced a list of keywords, concepts and associations that characterise the categories into which they grouped the research questions of articles submitted. This list demonstrates conclusively that the subareas of the discipline (organisational, behavioural and managerial issues) are well established and attract a large number of researchers on a long-term basis. Swanson & Ramiller conclude by observing that the discipline still exhibits the 'fragmented adhocracy' identified by Banville & Landry, and is still topically diverse and '...based on appeals to significantly different and partly incommensurate reference disciplines'.

Thus, fragmentation can have adverse effects – something that information systems researchers should be aware of. However, fragmentation of the discipline of information systems may be evident in the field for a very long time. As has been pointed out [17, 8], the discipline as a whole follows trends in information technology, and researchers tend to build their interests around new technology (e.g. the earlier interest in expert systems and decision support systems, and current interest in computer-supported co-operative work). As information technology evolves, so the research interests will follow these new directions. Although we may wish it were different, it remains a fact that information technology is still a major reference discipline of information systems, and will remain so as long as researchers struggle to separate the fundamental or common issues in different fields from the technological ones.

Clearly, then, information systems is internationally well-established as a flourishing discipline. In the Southern African context it is important that the discipline should not merely flourish but be seen to flourish. To this end, this editorial calls on academics and especially on practitioners to add your contributions, via a submission to SACJ.

References

[1] M. Alavi and P. Carlson. A review of mis research and disciplinary development. *Journal of Management Information Systems*, 8(4):45–62, 1992.

[2] R.L Ashenhurst. Curriculum recommendations for graduate professional programs in information systems. *Communications of the ACM*, 15(5):364–398, 1972.

[3] C. Banville and M. Landry. Can the field of mis be disciplined? *Communications of the ACM*, 32(1):48–60, 1989.

[4] H. Barki, S. Rivard, and J. Talbot. A keyword classification scheme for is research literature: An update. *MIS Quarterly*, pages 209–226, June 1993.

[5] R.B. Cooper and M. Blair, D.and Pao. Communicating mis research: A citation study of journal influence. *Information Processing & Management*, 29(1):113–127, 1993.

[6] J.L. Couger. Curriculum recommendations for undergraduate programs in information systems. *Communications of the ACM*, 16(12):727–749, 1973.

[7] J.L. et al. Couger. Guidelines for undergraduate is curriculum. *MIS Quarterly*, 19(3):341–360, 1995.

[8] P. Ein-Dor and E. Segev. A classification of information systems: Analysis and interpretation. *Information Systems Research*, 4(2):166–204, 1993.

[9] S.B. Eom and S.M. Lee. Leading us universities and most influential contributors in decision support systems research (1971-1989). *Decision Support Systems*, 9:237–244, 1993.

[10] J.T. et al. Gorgone. Information systems '95 curriculum model — a collaborative effort. *Data Base*, 25(4):5–8, 1994.

[11] R.A. Hirschheim, J. Iivari, and H. Klein. An assumption analysis of the five emergent approaches to information systems development (isd): A contribution to the paradigmatic foundations of alternative approaches to isd. Working paper, Binghamton, 1995.

[12] C.W. Holsapple, L.E. Johnson, H. Manakyan, and J. Tanner. Business computing research journals: A normalized citation analysis. *Journal of Management Information Systems*, 11(1):131–140, 1994.

[13] J. Iivari. A paradigmatic analysis of contemporary schools of is development. *European Journal of Information Systems*, 1(4):249–272, 1991.

[14] B. Ives. Posting to `listserv.hea.ieon` 21 July 1997.

[15] P.G.W. Keen. Mis research: Reference disciplines and a cumulative tradition. In *Proceedings of the First International Conference on Information Systems, Philadelphia*, pages 9–18, 1980.

[16] P.G.W. Keen. Relevance and rigor in information systems research: Improving quality, confidence, cohesion and impact. In H-E. Nissen, H.K. Klein, and R.A. Hirschheim, editors, *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland, Amsterdam, 1991.

[17] K. Lyytinen. Information systems and critical theory. In M. Alvesson and H. Willmott, editors, *Critical Management Studies*. Sage Publications, London, 1992.

[18] J. Nunamaker, J.L. Couger, and G.B. Davis. Information systems curriculum recommendations for the 80s: Undergraduate and graduate programs. *Communications of the ACM*, 25(11):781–805, 1982.

[19] E.B. Swanson and N.C. Ramiller. Information systems research thematics: Submissions to a new journal, 1987-1992. *Information Systems Research*, 4(4):299–330, 1993.

Fractal Image Compression

E. Cloete[a] L.M. Venter

[a]Dept. Computer Science and Information Systems, UNISA, Pretoria, 0001, cloete@alpha.unisa.ac.za
[b]Dept. Computer Science and Information Systems, PU for CHE, Vanderbijlpark, 1900

Abstract

Conventional image compression strategies present a number of uncomfortable limitations. Many of the impediments can be overcome by Fractal Image Compression (FIC). FIC is rooted strongly in mathematics. In comparison to traditional images compression strategies, FIC outperforms them by far when looking at compression ratios. FIC is thus an attractive compression scheme for the users and implementors of Multimedia and other graphical environments. Presently FIC has a number of irritating implementation difficulties, but since its foundation is sound, we believe it is a matter of research before these will be solved.
In this article we will give a brief description of the mathematical background involved in FIC, and show how the mathematics are used to achieve the compression. Our aim is to present the material in a form understandable to most scientists. We also discuss some of the open problems in the field.
Keywords: *Fractal Image Compression, contraction, IFS, self-similarity, transformations*
Computing Review Categories: *A.1, I.3.5, I.4.2*

1 Introduction

Traditional standards for graphical image compression have overcome many of the storage problems encountered when working with large graphical files, but some problematic limitations remain. Some of these limitations include:

- Traditional compression schemes are resolution dependent. With the fast growing computer industry, it is not desirable to be dependent on specific hardware.
- To achieve very high compression ratios, a drastic degradation in the decompressed version can be expected. For most applications, this is not acceptable.
- Since traditional compression schemes tend to be lossy, thus losing some information permanently, the loss becomes very evident when trying to zoom into a specific part of a decompressed image.

Most of these problems may be avoided by using Fractal Image Compression (FIC). This is an emerging technique which is not yet fully developed. It is based on mathematical theory which is simple to understand. However, there are still problems with the implementation of FIC.

2 History of Fractal Image Compression

The term *fractal* was first used by Mandelbrot [8] in the mid seventies, to describe objects that have irregular and fragmented shapes. These objects can be described in mathematical terms, not by an analytic formula but by a rather simple procedure. Mandelbrot did not actually consider fractals for compression but he pointed out that they could be used for modeling real world objects such as clouds, trees or mountains. In our setting, a fractal image is simply a set of points in the two-dimensional plane which exhibit a variety of detail, yet this set can be described by a special set of uncomplicated functions [9].

This can be viewed as an extreme form of compression: the algorithm itself can be described with a few bits of information or implemented in a very short program, but the resulting image would need a large number of bits to be represented as a set of pixels [12].

In the late 80's, Micheal Barnsley recognized the potential of fractal techniques for image compression. He foresaw compression ratios of up to 10 000 to 1 [2]. The process suggested by Barnsley [1] relies on the Collage theorem (see theorem 2 below), which states that an image may be approximated by using a special set of functions, the so-called Iterated Function System (IFS). The IFS is stored and later used to reproduce the image, without any additional input like dictionaries, lookup tables or code books. The process was patented by Barnsley and Sloan [3].

The process to go from image to IFS is known as the inverse problem. Finding such an IFS for an image was first automated in 1988 by Arnaud Jacquin, a student of Barnsley, who showed that it is practically impossible to find an IFS for an entire image. Instead, he partitions the image into a number of subimages, prescribes a method to find an IFS for each partition, and in the end uses the collage of all the partitioned IFS's to regenerate the original image [5]. This process is known as the fractal compression with PIFS (*Partitioned Iterated Function System*).

3 Theoretical Background

An Iterated Function System

Fractal image compression is based on the following principle: instead of representing the image as a long sequence of pixel values, the image can be reconstruced from the special set of functions, which can be encoded in less bytes [12]. The following NON-FRACTAL example can be used to illustrate the point.

If we have an image which consist of a black ellipse on a white background, we do not need to store the image pixel by pixel, we merely need to know the center of the ellipse and its formula. We store only the formula, its center point and colour representation, and may then reconstruct the image precisely as it was using a standard plotting algorithm on the input formula. Essentially fractal image compression makes use of the same principals, although the images cannot in general be described by a single analytical formula.

To illustrate the fundamentals of this technique, we define a number of concepts. The interested reader can consult Barnsley [1] and Kreyszig [6] for a further explanation of the mathematical terms used. All of the mathematics involved is done in the *metric space* setting. A metric space (X,d) is simply a set X and a function d that maps the Cartesian product $X \times X$ into the non-negative real line, i.e. $d : X \times X \longrightarrow \mathbb{R}^+$. This function has a number of (specified) properties. Geometrically, it is simply a function that measures distances in X. In the case when the set X is the two-dimensional plane $\mathbb{R}^2$, the function d can be the usual Euclidean distance $d(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$.

Definition 1 *Contractive transformation [1]*
A transformation $w : X \longrightarrow X$ on a metric space (X,d) is called contractive *or a* contraction mapping *if there is a constant $0 \leq s < 1$ such that*

$$d(w(x),w(y)) \leq s.d(x,y) \forall x,y \in X,$$

and s is called the contractivity factor *for w.*

This definition says that a transformation is contractive it if reduces distances.

Theorem 1 *The Contractive Mapping Theorem [1]*
Let $w : X \longrightarrow X$ be a contraction on a complete metric space (X,d). Then, there exists a unique point $x_f \in X$ such that $w(x_f) = x_f$. Futhermore, for any $x \in X$, we have

$$\lim_{n\to\infty} w^{0n}(x) = x_f,$$

where w^{0n} denotes the n-fold composition of w with itself. The point x_f is called the fixed point *of w.*

The important point of the theorem is that every contractive mapping w has a unique fixed point, that is, a value x such that $w(x) = x$.

Definition 2 *Iterated Function System (IFS) [1]*
A (hyperbolic) iterated function system consists of a complete metric space (X,d) together with a finite set of contraction mappings $w_n : X \to X$, with respective contractivity factors s_n, for $n = 1,2,\ldots,N$. We denote the IFS by $\{(X;w_n), n = 1,2,...,N\}$ and its contractivity factor is $s = \max\{s_n : n = 1,2,...,N\}$

Hence, an IFS is simply a finite set of contractive mappings.

Consider a metric space (X,d) and a finite set of N strictly contractive transformations $w_i : X \to X, 1 \leq i \leq N$, with respective contractivity factors s_i i.e. an IFS. Furthermore, let $\mathcal{H}(X)$ be the set of all nonempty, compact subsets of X. With the *Hausdorff metric*, defined by

$$h(A,B) = \max\{\min\{d(x,y) : y \in B\} : x \in A\}$$

$\mathcal{H}(X)$ is also a complete matric space. Define a transformation $W : \mathcal{H}(X) \to \mathcal{H}(X)$, by

$$W(B) = \bigcup_{i=1}^{N} w_i(B)$$

for any $B \in \mathcal{H}(X)$. It follows from the contraction mapping theorem that W has a unique fixed point A in $\mathcal{H}(X)$, satisfying the condition

$$A = W(A) = \bigcup_{i=1}^{N} w_i(A).$$

This unique fixed point is called the *attractor* or *fractal* of W.

We are now ready to formulate the Collage Theorem.

Theorem 2 *Collage Theorem*
Let (X,d) be a complete metric space and choose $I \in \mathcal{H}(X)$. Fix $\varepsilon \geq 0$. If $\{X;w_0,w_1,w_2,...,w_n\}$ is an IFS with contractivity factor $0 \leq s < 1$ and fixed point $A \in \mathcal{H}(X)$ such that

$$h(I,W(I)) \leq \varepsilon,$$

then it follows that

$$h(I,A) \leq \frac{\varepsilon}{(1-s)}.$$

The Collage Theorem promises that if an IFS can be found which, when applied to the original image, produces a very close approximation of the original image, then the fixed point of the IFS will also be a good approximation of the image. This implies that we can store the IFS and then when decompression is necessary, apply this IFS repeatedly to **any image** to produce an approximation of the original image.

Although some researchers [4] have demonstrated that an iterated function system for an arbitrary image can be found, the main thrust of research is aimed at how to find the IFS.

Images usually contain a type of self-similarity, in that regions of an image are similar at different scales. That means that such an image can be reconstructed by transforming *parts* of itself. Such a recreation process is demonstrated by the *photocopy machine algorithm* which shows how the image is formed by copies of its *whole* self under specific affine transformations. The example leads to the discussion of an encoding algorithm to construct the affine transformations of an image.

Example 1 - Explaining self-similarity In the photocopy machine algorithm, an initial image is shrunk, pasted and copied. This procedure is applied several times until no visible changes from one iteration to the next can be observed. As an example, let us start with three triangles placed in triangular form. Make a copy of this picture and shrink the picture to the size of one triangle in the prior picture. Make enough copies of the deflated image and paste a copy on each of the original triangles. This process is repeated and after about 10 iterations, the visual appearance of the image do not change. The resulting image is known as the Sierpinski triangle.

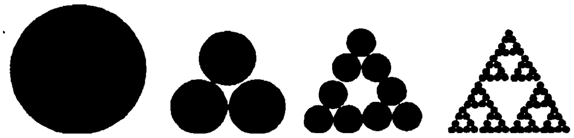

Figure 1: The first four iterations of the Sierpinsky Triangle through self-similarity

An encoding approach to find an IFS

Before beginning our discussion, we need one more definition.

Definition 3 *Affine Transformation*

A transformation $w : \mathbb{R}^n \rightarrow \mathbb{R}^n$ *of the form* $w(x) = Ax + b$ *for any* $x \in \mathbb{R}^n$ *is called* affine. *A is an* $n \times n$ *matrix and* b *is an n-dimensional vector.*

Affine transformations are often used in graphical representation to allow for scaling, rotation and reflection by matrix multiplication with **A** and a translation with **b**.

Most of the existing fractal compression algorithms are based on the principles defined by Jacquin. Different algorithms differ in certain steps. Essentially, these are all brute-force methods to construct an IFS. Fractal compression with PIFS works by finding redundancy within the input image in the form of similar image portions.

In his approach Jacquin [5] makes use of the work of Ramamurti & Gersho [13] who have defined specific partitioning classes according to which they suggest one can classify image blocks. Their work deal mainly with image processing. The partition classes they use are:

- *shady blocks* (blocks which have no luminance variance);
- *midrange blocks* (blocks with a luminance variance less than 2%);
- *edge blocks* (block where there is a definite luminance variance). The edge blocks can be further subdivided into:
    1. *simple edge blocks* (for blocks where the edge goes horizontally, vertically or diagonally through the block)
    2. *mixed edge blocks* (for all other edge blocks)

Each of these classes of partitions is called a cell pool.

According to Jacquin's work [5,6] the encoding process consists of distinct steps which can be described by the fractal encoder as illustrated in the following Figure 2.

Step 1: *Partition the image support into range cells.* The support of the image is a square frame around the image to provide borders to work in. Partition the support into $B \times B$ range blocks (usually 4×4 pixels squares). These blocks may not overlap. Each range cell is now classified into a specific cell pool.

Step 2: *Partition the image support into domain cells.* Partition the support into $D \times D$ domain cells (usually twice the size of a range cell). These blocks overlap and provide many domain cells used in the process of defining a transformation for a specific range block. Domain cells need not be classified at this stage.

The principal idea of Jacquin's encoding process is to reproduced each range block by a single contraction. Although theorem 1 states that one can start with any image in the search for the IFS, the domain blocks seem to be a good set to experiment with for finding this transformation.

Step 3: *Define transformations from domain to range cells.*

Each range cell is subject individually to step 3 and step 4. Each contractive transformation defined in this step is composed of two individual transformations, namely a *spatial transformation* and a *massic transformation.*

After the source image has been partitioned into domains, each domain cell is immediately deflated to the size of the range blocks. This is the first part of the contractive mapping, the *spatial transformation.* These deflated domain cells are then classified into domain pools according to their pixel properties. To find a transformation reproducing a specific range block, all the domain cells with similar classification as the particular range cell, are considered.

The second map in the contractive transformation is called the *massic transformation.* The massic transformations are really the affine transformations of the encoding process. Here, a domain block is translated, reflected and/or rotated until it fits onto a particular range block. The

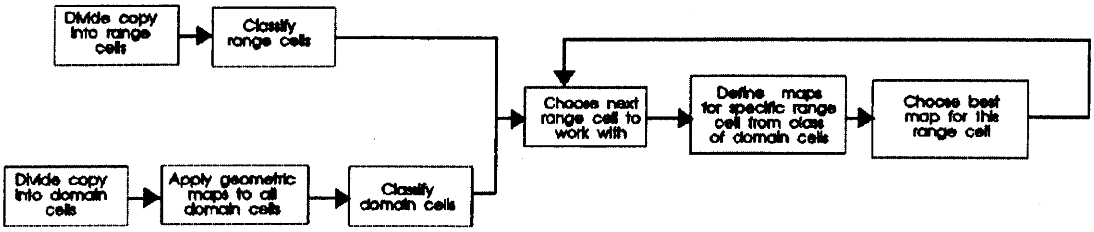

Figure 2: The Fractal Image encoder

process, and not the block itself, which a domain block undergoes, is saved as the affine transformation.

Before we discuss these transformations, let us define the following notations to simplify the reading of this paragraph:

R_i refers to the i^{th} range block.

D_j refers to the j^{th} domain block in a specific domain pool, where $j = 0 \ldots$ *number of domain blocks in this domain pool.*

$\rho(D_j)$ refers to the domain cell, D_j, which has been contracted by the spatial transformation, ρ. For the rest of this paragraph we will simply donate $\rho(D_j)$ as ρ_j.

Spatial Transformations: The spatial transformations are rather simple in that it simply shrinks all $D \times D$ domain block into $B \times B$ sizes. If $D = 2B$, this can be accomplished by taking the average of 4 neighbouring pixel values as the new pixel value.

Massic Transformations: In order to find a good approximation of the range block R_i, we need to define a massic transformation, Γ_j, on each contracted domain block ρ_j, from which the best massic transformation can finally be selected. Jacquin defines a separate type of massic transformation for each domain pool.

- **Massic Transformation for the Shady Domain Pool**
  This transformation merely scales the pixel values in the domain block in such a way that the average value matches that of a particular range block.

- **Massic Transformation for the Midrange Domain Pool**
  Transformations on the *midrange* domain pool are of the form:
  $$\Gamma_i(\rho_j) = \alpha_i(\rho_j) + \xi_i$$
  where
  - α_i is a constant contrast scaling value. The number of bits needed to store the transformation can be succesfully reduced in this step by performing *quantization*. For this transformation, α_i is the quantization factor used to quantize the matrix values of ρ_j.
  - ξ_i is called the brightness and is calculated as the quantized difference between the average pixel values in the specific range block and domain block: $\xi_i = \overline{R_i} - \alpha_i \overline{D_j}$

- **Massic Transformation for the Edge Domain Pool**
  Transformations on the *edge* domain pool are of the form
  $$\Gamma_i(\rho_j) = l_{ni}(\alpha_i(\rho_j) + \xi_i)$$
  where
  - α_i is not merely a constant value as in the case of the Midrange domain pool, but a variable contrast scaling value which also performs quantization. Since we are considering blocks with a sharp greylevel step, it is necessary to reflect this step as well as the dominant greylevel of the particular block. Each pair (R_i, ρ_j) needs to be analyzed. This is done via further segmentation of each block in the pair. A histogram is set up for each block to determine a threshold between the two distinct greylevel batches. In this way, the dominant region of each block is identified. A dynamic range, dr, can now be calculated for each block. The dynamic range is merely the greylevel difference of such a segmented block. Finally $\alpha_i = min\{(dr(R_i)/dr(\rho_j)), \alpha_{max}\}$ where α_{max} is a fixed maximum value.
  - The brightness, ξ_i, is computed so that the dominant greylevel in the particular segmented blocks have the same intensity.
  $$\xi_i = Dominant(R_i) - \alpha_{max} Dominant(D_j)$$
  - l_{ni} refers to the i^{th} range block for which an isometry, l, is selected. There are a number of possible isometries (eg. reflection, rotation, etc) but in this case 8 isometries are defined, therefore $0 \leq n \leq 7$.
  The isometry which minimizes the distortion measure is selected. This implies that for each possible transformation, seven more transformations (one for each isometry except the identity isometry) are calculated, before the best one can be selected.

Step 4: *Select the contractive transformation which will provide the best approximation for the current range block*
In the massic transformation step several transformations have been defined which may each possibly recreate a particular range block. Together with each massic transformation, we need to define a deviation error, also known as the

Table 1: IFS for Sierpinski triangle

Maps	a	b	c	d	e	f
w_1	0.5	0.0	0.5	0.0	0.5	0.0
w_2	0.5	0.0	0.5	0.0	0.5	0.5
w_3	0.5	0.0	0.5	0.0	0.25	0.5

distortion measure, which describes how "close" the approximation is to the real image block. The range block R_i, is approximated by a particular transformation and recreated as the image segment S_{ij}.

The "difference" between R_i and S_{ij}, is defined by the distortion measure, $d(R_i, S_{ij})$ as:

$$d(R_i,S_{ij}) = \sqrt{\sum_{0\leq p,q<B} (\mu_{p_0+p,q_0+q} - S_{p_0+p,q_0+q})^2}$$

- S_{ij} refers to the image generated by the tranformation from the j^{th} domain block onto the i^{th} range block.
- μ_{p_0+p,q_0+q} is the $(p,q)^{th}$ element of R_i
- ν_{p_0+p,q_0+q} is the $(p,q)^{th}$ element of S_{ij}

The transformation which offers a distortion measure value as close as possible to 0, when compared to the original image block, provides the best possible affine map. Since the encoding procedure is so time consuming, a threshold is usually defined, and the first transformation which offers a distortion measure below this threshold is chosen as *the* affine map to be saved, and the others are discarded. This speeds up the encoding algorithm slightly but may not necessarily be the transformation which offers the highest compression ratio.

Example 2 - The transformations for the Sierpinski triangle The photo-copy algorithm discussed earlier forms a good basis for FIC since the Sierpinski triangle image exhibits self-similarity. Example 1 shows how parts of the picture is taken, transformed in some way and used again to produce the final image. Performing this task manually on a photo-copy machine is impracticle. In an interactive multimedia environment where images have to be compressed and decompressed online, an automated procedure is needed. The Sierpinski triangle displays all the properties to be an excellent candidate for the fractal image compression scheme. When the encoding procedure is applied to the Sierpinski triangle, only three transformations are found such as displayed in Table 1.

Each transformation is in the form $w_k(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ where $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is a 2×2 matrix and $\mathbf{b} = \begin{bmatrix} e \\ f \end{bmatrix}$ is a 2-dimensional vector.

These transformations describe exactly the activities performed by the photo-copy algorithm. The question may arise how the affine transformation $w_k(\mathbf{x}) = A\mathbf{x} + \mathbf{x}$, discussed here, relates to the affine transformation $\Gamma_i(\rho_j) = l_{ni}(\alpha_i(\rho_j) + \xi_i)$, given by Jacquin. To discuss this question, we can think of an image in the metric space of digital images as a matrix consisting of a number of "pixels". Each of these pixels can be uniquely described by a pair (P,z) where P describes its position in the picture while z describes its "colour" or greyscale level. In the simple case of the photo-copy algorithm, we agree that P will change, but not z. In the Jacquin-process, we keep P fixed and change z. Hence, when we write $\Gamma_i(\rho_j) = \alpha_i(\rho_j) + \xi_i$ we mean that for each pixel $(P,z) \in \rho_j$ the following calculation is done:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \alpha_i \end{array}\right] \begin{bmatrix} \mathbf{P} \\ z \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \xi_i \end{bmatrix}$$

which is exactly the form $A\mathbf{x} + \mathbf{b}$

4 Fractal Decompression

The decompression algorithm is based on iterating the IFS until the sought after image is obtained. To perform one iteration of the PIFS, the algoritm takes the list of all affine maps and applies each one in turn on an initial image (say a black square). This transforms a set of domains into a set of ranges. The whole process is repeated, this time on the image obtained after the first iteration. The process is repeated until there is very little difference between the input image and the output image, usually within 8 iterations [12].

If the initial image is not of the same size as the original image, the resulting image will have the same size as that of the initial image and not of the original image. Scaling the image is thus an easy process with fractal image compression [12].

5 Problems with Fractal Image Compression

- *Artificial Information.* Taking any original image, say a face, and zooming into it excessively, the resulting image shows the pores in the skin. This does not necessarily happen with fractal image compression. One might only see a very smooth skin.
- *Patented methods hides knowledge from the research world.* The field of data compression is mined with patents, and fractal image compression is no exception [12]. Most fractal patents are owned by Iterated Systems, Inc (Company of Barnsley and Sloan).
- *Encoding algorithm of the inverse problem.* The encoding algorithms presently available, are extremely slow when comparing them to the encoding algorithms offered by traditional compression schemes.
- *Image Partitioning.* Jacquin makes use of the quadtree block partitioning scheme in which he simply slices the image support into a number of smaller blocks.

This partitioning algorithm is based on the observation that a real-word image contains self-similar parts. Since there is no or very little "intelligent decision" in this segmentation method, it may not provide the best compression ratios. It also leaves the search for finding the best transformation with almost an uncountable number of transformations to define and compare for every range block. Other researchers such as Lu & Yew [7],

Jacquin [5] have suggested, and Monro [11] and McGregor [10] have shown, that that one does not necessarily have to partition the image into square cells. Better compression results may be achieved by using different rectangles or even triangles in this partitioning. Other partitioning forms may however increase complexity and calculation time.

- *Classification of the Image.* Test results based on Jacquin's approach concluded that the quality of the decoded fractal image relies heavily on block classification and analysis. Transformations defined on edge or shady blocks preserves data fairly accurately, but the midrange blocks which allows for a very wide set of greylevel changes, shows degradation in the image.

6 Conclusion

The decompression algorithm of fractal image compression is currently faster than the traditional compression standards for graphical images [12]. The encoding algorithm however is slow and hence leaves fractal compression as a potential standard and not as a standard.

References

[1] M F Barnsley and L P Hurd. *Fractal Image Compression.* A K Peters, Wellesley, Massachusetts, 1993.

[2] M F Barnsley and A D Sloan. 'A better way to compress images'. *Byte*, pp. 215–223, (January 1988).

[3] M F Barnsley and A D Sloan. Method and apparatus for image compression. United States Patent # 5065447, 1991.

[4] G A Edgar. *Measure, Topology and Fractal Geometry.* Springer-Verlag, New York, 1990.

[5] A Jacquin. 'Image coding based on a fractal theory of iterated contractive image transformations'. *IEEE Transactions on image processing*, **1**(1):18–30, (January 1992).

[6] E Kreyszig. *Introductory Functional Analysis with Applications.* John Wiley & Sons, New York, 1978.

[7] G Lu and T L Yew. 'Image compression using quadtree partitioned iterated function systems'. *Electronic Letters*, **30**(1):23–24, (January 1994).

[8] B B Mandelbrot. *Les objets fractals: forme, hasard et dimension.* Flammarion, Paris, 1975.

[9] B B Mandelbrot. *The fractal geometry of nature.* W H Freeman and Company, New York, 1977.

[10] D R McGregor, R J Fryer, P Cockshott, and P Murray. 'Faster fractal compression'. *Dr Dobb's Journal*, pp. 34–40, (January 1996).

[11] D M Monro and F Dudbrigde. 'Fractal approximation of image blocks'. *Proc IEEE Conference on Acoustics, Speech and Signal Processing*, **III**:485–488, (March 1992).

[12] M Nelson. *The Data Compression Book.* M & T Books, New York, 1996.

[13] B Ramamurthi and A Gersho. 'Classified vector quantization of images'. *IEEE Transaction Communications*, **COM-34**, (Nov 1986).

Received: 11/96, Accepted: 7/97

Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications of Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript

Manuscripts for *review* should be prepared according to the following guidelines:

- Use wide margins and $1\frac{1}{2}$ or double spacing.
- The first page should include:
- the title (as brief as possible)
- the author's initials and surname
- the author's affiliation and address
- an abstract of less than 200 words
- an appropriate keyword list
- a list of relevant Computing Review Categories
- Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text according to the Harvard. References should also be according to the Harvard method.

Manuscripts accepted for publication should comply with guidelines as set out on the SACJ web page,

`http://www.cs.up.ac.za/sacj`

which gives a number of examples.

SACJ is produced using the LaTeX document preparation system, in particular LaTeX 2_ε. Previous versions were produced using a style file for a much older version of LaTeX, which is no longer supported. Please see the web site for further information on how to produce manuscripts which have been accepted for publication.

Authors of accepted publications will be required to sign a copyright transfer form.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30.00 per final page for contributions which require no further attention. The maximum is R120.00, prices inclusive of VAT.

These charges may be waived upon request of the author and the discretion of the editor.

Proofs

Proofs of accepted papers may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words. Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book Reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement

Placement of advertisements at R1000.00 per full page per issue and R500.00 per half page per issue will be considered. These charges exclude specialised production costs, which will be borne by the advertiser. Enquiries should be directed to the editor.

South African Computer Journal

Number 21, August 1998
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 21, Augustus 1998
ISSN 1015-7999

Contents