J M Bishop
20/3/90

South African Computer Journal
Number 1
January 1990

Suid-Afrikaanse Rekenaar-tydskrif
Nommer 1
Januarie 1990

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

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

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

Subscriptions

	Annual	Single copy
Southern Africa:	R32-00	R8-00
Elsewhere:	\$32-00	\$8-00

to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Editorial

At last the first edition of SACJ is available. I trust that readers will find it worth the waiting. There have been a number of teething problems in getting things together, the many details of which need not be spelt out here. One significant challenge was to cope with the consequences of the resignation of Quintin Gee, QI's highly competent production editor. He assisted in the initial phases of getting this publication together but had to resign for personal reasons. It is fitting to acknowledge here not only his initial advice and assistance in getting this first issue of SACJ off the ground, but also the many hours of work that he spent in previously producing QI.

Quintin's resignation meant that a new *modus operandi* for typesetting and printing had to be established. The exercise was not only time-consuming, but also has significant cost implications. Fortunately, the Unit for Software Engineering (USE) at Pretoria University has generously agreed to sponsor this first edition. On behalf of the South African computing community, I should like to thank them for their generosity. Now that they have made a first issue of SACJ possible, it is hoped to solicit the sponsorship of one of the larger computer companies for future editions.

It might be of interest to take readers on a walk through the new journal to highlight various aspects. To begin with, the cover design follows that of several journals whose titles have the format: *The South African Journal of Subject / Die Suid-Afrikaanse tydskrif vir Vakgebied* (where *Subject* and *Vakgebied* are appropriately instantiated). While colours vary, these journals generally have *Subject* and *Vakgebied* restated on the darker portion of the cover. SACJ's title was chosen in preference to a more descriptive but also more cumbersome title such as *The South African Journal of Computer Science and Information Systems*. The appearance of the words *Computer Science and Information Systems / Rekenaarwetenskap en Inligtingstelsels* on the cover are thus out of step with the original inspiration, but seem appropriate under the circumstances.

The inside cover is of interest for several reasons. Firstly, note that Peter Lay has kindly agreed to lighten my task by acting as an assistant editor. He will deal with matters relating to Information Systems. *Contributions in this area should henceforth please be sent directly to him.* Also note that an editorial board of distinguished persons has been assembled. I should like to once again thank board members for adding status to SACJ by agreeing to serve in this capacity. They will be consulted on matters of editorial policy whenever appropriate. Finally, the subscription costs have been increased to keep pace with production costs. This increase does not affect SAICS members, who will continue to receive the journal as one of the benefits of membership.

The guest editorial by Pieter Kritzinger makes for interesting reading. Several points of concern about computer-related research in South Africa are raised. I trust that the article will focus attention on these problems and stimulate a debate which will lead to eventual solutions. It is hoped to make guest editorials a regular feature of future SACJ issues.

Of the eight research papers offered in the journal, four have been gone through the normal channel of refereeing and revisions. The remainder were submitted to the Vth SA Computer Symposium and are published here by invitation. Each paper submitted to the chairman of the symposium's program committee was sent to three referees. A ranking scheme, reflecting an aggregate measure of referee evaluation, was used as a basis for deciding on papers to be presented. After further editorial evaluation, the authors of four of the five highest ranking papers were invited to submit their papers to SACJ. While it was not possible to contact the fifth author in time for this edition, but it may be possible to publish that paper, together with a selection of others from the symposium, in future SACJ editions.

In the section marked *Communications* various items of news arriving at the editor's desk have been published. It was particularly gratifying to receive book review submissions in response to a prior general appeal. There has also been an enthusiastic response from book publishers, who have sent in a number of books for review. Titles are listed in the *Communications* section. Please contact me if you are willing to review one (or more) of these. Naturally, reviews of other books of interest in your possession will also be welcomed.

The final point to highlight in this walk through the journal is the increase in page charges indicated on the back inside cover. These reflect the increased cost of production. Since research papers in SACJ qualify for state subsidy at academic institutions, the charges should not, in general, present major problems for authors. However, it is worth pointing out that the final format of papers submitted significantly impacts on both the financial and editorial load. Submissions in camera-ready format (or nearly so) result in both a cost savings and a speed up of turn-around time by several orders of magnitude. Since many readers may not be familiar with the printing process, it may be helpful to say something about it in order to substantiate this claim.

The printing process basically involves typesetting, shooting (or photographing), and then reproduction and binding. Apart from limiting the amount of material, the printer's client has very little control over the cost of shooting, reproduction and binding. On the other hand, anyone equipped with moderate text- or word processing facilities and a laser printer can go a long way (if not all the way) towards typesetting a paper. Even a partially typeset paper helps significantly, as I will explain below.

By typesetting I simply mean knocking the paper into the right shape and producing a laser printout. The printers regard this is a tedious, error-prone task, even if they start off with an ASCII file rather than a hardcopy of the paper. Consequently, they tend to handle large-scale typesetting by subcontracting the task. Moreover, while they may be willing to typeset uncomplicated text, they tend to balk at text containing specialized mathematical and other notation. However, they are quite skilful at cutting and pasting text, and at enlarging or reducing photographed or scanned diagrams. They are even willing to redraw sketches which are not too complicated.

As a result of the above, I have pressed several authors to do their own typesetting. In cases where it was problematic to produce double column format, a single column of appropriate width was requested. While this is a second-best option, it allows for cutting and pasting to be done by the printers. Some sketches have either been directly reduced from the author's original, while others have been redrawn by the printer. By way of exception, I have personally undertaken the typesetting of a few papers using WordPerfect. However, I would like to avoid this as far as possible in future, and consequently appeal to potential authors to make every effort to do their own typesetting.

From SACJ's point of view encouraging authors to do their own typesetting involves a compromise in that there will inevitably be slight variations in the print from one article to the next (as is in fact the case in this issue). If you are pedantically inclined, you might consider this to be a disaster. Personally, I regard it as a rather neat advertisement for the typesetting skills of SACJ contributors.

As an aside, since the handling of $\TeX$ files was initially a problem for me, I was pleased to discover that Peter Wood and his colleagues at UCT have mastered the art of producing $\TeX$ printout in the format now before you. Future authors who use $\TeX$ should consult them on details.

As to the future, it is not possible at the this stage to commit to a fixed number of SACJ issues per year. The number of issues is constrained by finance, submissions of the right quality, and time available to the editorial staff (including our anonymous and unsung heroes - the referees). The ideal is to produce four issues per year, but this may not always be attainable.

In conclusion, if readers have as much fun in reading this first issue of SACJ as I have had in editing it, the hours spent on it will have been well worthwhile. Hopefully SACJ is destined not only to be a permanent feature of the Southern African computing scene, but also to significantly contribute to research in the region.

Derrick Kourie
Editor

This SACJ issue is sponsored by
The Unit for Software Engineering
(USE)
Department of Computer Science
Pretoria University

Funding Computer Science Research in South Africa

P S Kritzinger
Department of Computer Science, University of Cape Town, Rondebosch 7700

The word *research* has many connotations and is often abused. In everyday language a person does not simply *search for information in a library*, for example, but rather does *research*, thus pretentiously conferring an aura of intellectual activity on an effort which requires very little original thought.

Here I will interpret the term to mean work which generates results that gain international recognition. This implies that the work is published in good international journals or presented at international conferences. I believe this is the only valid index of the quality of research.

With very few exceptions, the computer industry in South Africa is a consumer of computer technology, rather than a developer. In contrast with, say, the chemical industry, there is therefore no tradition of research in computer science in the South African computer industry and computer science researchers therefore have, as virtually their only source of funding, the Foundation for Research Development (FRD) which has its origins in the CSIR.

The FRD was formed in April 1984 with the development and use of research expertise in the natural and applied sciences and engineering as its mission. This mission is primarily directed at the universities, museums and technicons with the ultimate aim of improving the life of all South Africans.

Although the FRD has several programmes, the two which are of main concern to computer scientists are the Core Programmes and the Special Programmes.

FRD Core Programmes foster the optimum development of a scientific and technological knowledge base by supporting individual self-initiated research. These programmes, started only about 4 years ago, have met with considerable acclaim, particularly in regard to the way in which research funding for a particular individual is decided. To qualify for support within a Core Programme, researchers must obtain a certain evaluation status within the FRD and funding is then linked directly and exponentially to the merit of the individual concerned, rather than being linked to the specific project proposed.

In the evaluation process, peer review is strongly emphasised. The researcher himself is expected to nominate referees, whose status and reports play a decisive role in the evaluation. As a result of this evaluation, an applicant is assigned a specific evaluation status category. There are currently 9 categories in all, but the ones of main interest are:

- **A** researchers who are without any doubt accepted by the international community as being amongst the leaders in their field (52);
- **B** researchers not in category A but who nevertheless enjoy considerable international recognition as independent researchers of high quality (182);
- **C** proven researchers who have maintained a constant high level of research productivity and whose work is regularly made known internationally, or proven researchers whose current research output is less but who are actively engaged in scholastic activity (433);
- **P** researchers younger than 35 years of age who have already obtained a doctoral degree and who have shown exceptional potential as researchers (10); and
- **Y** young researchers usually under 35 years of age, who are highly likely to achieve C status by the end of their support period (108).

The number of researchers in the various categories as of August 1989 has been indicated in parentheses above. Of these, only 7 persons are computer scientists: 1 in category B; 3 in category C; and 3 in category Y. Only 4 departments of computer science are involved.

The other main programmes of concern to computer persons are the Special Programmes which aim at developing research manpower in priority areas. After identification of an area that merits particular research development, given local expertise, a Special Programme is launched to address the problem in the national interest.

Although a manager of a Special Programme has to be an FRD evaluated researcher, the same need not be true for the other team members. Regular peer evaluation of researchers as well as evaluation of the progress and results of Special Programmes are considered essential. Special Programme awards will be made for the first time towards the end of 1989. It is therefore not yet known whether proposals already submitted for programmes in computer science have been successful.

It is clear that, in the context explained above, there is virtually no computer science research being done in South Africa - a scary thought which has considerable implications for this country! Why is this so? There are several reasons, but I would like to single out two in particular.

Qualified faculty and students is an abiding problem at the heart of computer science departments. Acquisition of new faculty members is an issue intimately linked to the number of graduate students successfully completing PhD degrees. This problem is by no means unique to South Africa. For instance, data gathered in North America indicates that in 1983 there were over 200 vacancies in the 91 departments that have doctoral programmes in computer science. At the same time, only approximately 250 PhD's were granted in North America - a figure that has remained relatively unchanged for the past several years. A large number of those graduates were attracted to industry and industrial research laboratories. Although I do not have solid data at my disposal, I would think that South Africa produces at most one PhD graduate in computer science per year. There are currently 20 departments of computer science at universities is South Africa. It will therefore take us 20 years to locally produce one new faculty member with a PhD in computer science for every university.

Contributing to the above problem is our current academic image. The graduate student usually sees concerned computer science faculty members as rather harried individuals, having large undergraduate classes, much committee and professional work, and labouring under an ill-fitting model (applicable to more established disciplines) for decisions on tenure, salary and promotion. Further, as undergraduates, many prospective graduate students were not engaged in research projects involving computer science faculty, and for that reason were not exposed to graduate students doing research, and rarely developed a camaraderie with any computer science professionals. At last count there were only 5 individuals in South Africa who completed their computer science doctorate at a university outside South Africa where they had the good fortune to work in an environment in which sufficient faculty and funds were available to create an ethos of research. It is difficult to convince students that their interests and goals can be served by a PhD in computer science or by an academic career.

The second problem, which is of greater concern to me since there is no immediate solution to it, has to do with the fact that senior persons who decide the fate and fortune of academic computer science departments are, in general, individuals whose professional careers started well before computing machines came into every day use - that is to say, in the years B.C. (Before Computers). These persons of influence do not always understand what "computers" are, and what their potential influence upon the workplace in particular and society in general are. As far as research (as opposed to teaching) is concerned, most of them understand that a medical school needs special and expensive equipment (not to mention, expensive faculty) and that engineers must have a workshop and special machinery to teach their students and conduct research. They understand that if one needs to build up a defense industry, it will cost billions of rands; but they are not so sure about computer science, even though many other countries have recognised it as of national strategic importance.

I believe that only time and dedication will lead to a solution of these seemingly insurmountable problems and allow computer scientists to take their rightful place in the research community in South Africa.

Bibliography

P J Denning, [1981], Eating our seed corn, *Communications of the A.C.M.*, **24**(6), pp.341 - 343.

J Tartar (Ed.), [1985], The 1984 Snowbird Report: Future issues in computer science, *Communications of the ACM*, **28**(5), pp490 - 493.

D Gries, R Miller, R Ritchie and P Young, [1986], Imbalance between growth and funding in academic computing science: two trends colliding, *Communications of the A.C.M.*, **29**(9), pp.870 - 878.

J E Hopcroft and D B Krafft, [1987], Towards better computer science, *IEEE Spectrum*, pp.58 - 60.

Image reconstruction via the Hartley transform

H C Murrell and D Carson

Department of Computer Science, University of Natal (Durban), King George V avenue, Durban

Abstract

The continuous and discrete Hartley transforms are real valued transforms that have similar properties to the continuous and discrete Fourier transforms.In addition, a fast algorithm exists for computing the discrete Hartley transform which is faster than the fast Fourier transform, even when the fast Fourier transform is optimized for dealing with real data.In this paper the authors apply the Hartley transform to the problem of image reconstruction.The authors will show that the projection-slice theorem and the filtered back-projection algorithm can be derived using the Hartley transform and that the filter part of the filtered back-projection algorithm can be implemented using Bracewell's fast Hartley transform.

Keywords: *reconstruction, tomography, Hartley transform.*
CR category: *I.4.5 (image reconstruction)*

Received January 1989, Accepted October 1989

1. Introduction:

The Hartley transform is named after R.V.L. Hartley who first introduced the transform in [3]. Bracewell [1] investigated the Hartley transform, and developed a decimation in time fast Hartley transform (FHT) for computing the discrete Hartley transform. A faster decimation in frequency FHT was developed by Meckelburg and Lipka in [5]. In particular, Meckelburg and Lipka note that an N point FHT is faster than an N point real FFT performed as an N/2 point complex FFT. Given any problem, Bracewell [1] shows that any solution based on Fourier transform techniques can be rewritten using the Hartley transform. If the problem involves real data then the new solution will be better than the old one since the FHT is faster than the FFT. In this paper we show that these ideas of Bracewell can be applied to the image reconstruction problem and the result is a faster algorithm for computer aided tomography. In section 2 we repeat Bracewell's definition of the Hartley transform. Section 3 deals with reconstruction theory in Kak's framework [4]. We show that the projection—slice theorem and the filtered back—projection algorithm have Hartley equivalents. In section 4 we state and prove a Hartley equivalent of the sampling theorem for band—limited functions. In section 5 we show that the filter part of the reconstruction algorithm can be discretized as a convolution. In section 6 we use [4] to repeat Kak's discretization of the back—projection part of the reconstruction algorithm. In section 7 we show how to compute the reconstruction convolution using a discrete Hartley transform and in section 8 we outline our testing procedures and give timings to support the claim that our new FHT based reconstruction algorithm is faster than the well known FFT based algorithm. Our contribution to the subject of reconstruction may be found in sections 3, 4, 5, 7 and 8. Sections 2 and 6 appear for the sake of completeness.

2. The Hartley transform:

In this section we give Bracewell's definition for the Hartley tansform and we include without proof the inversion theorem and convolution property for the Hartley transform. Many more Hartley transform theorems and properties with accompaning proofs can be found in Bracewell's book [1].

<u>Definition:</u> If h(t) is a real valued function of a real variable, then the Hartley transform of h(t) is also a real valued functon of a real variable, $H(\omega)$, defined by:

$$(2.1) \quad H(\omega) = \int_{-\infty}^{\infty} h(t)\ \mathrm{cas}(2\pi\omega t)\ dt$$

where

$$(2.2) \quad \mathrm{cas}\theta = \cos\theta + \sin\theta.$$

The usual provisos are made concerning the existance of the integral in eqn (2.1).

Inversion theorem: h(t) may be recovered from H(ω) according to :

$$(2.3) \quad h(t) = \int_{-\infty}^{\infty} H(\omega)\ cas(2\pi\omega t)\ d\omega$$

Note the reciprocal nature of Hartley inversion and that the kernel is real valued. An analogous transform with inversion theorem exists for functions of two variables. See [1] for details.

Convolution theorem: Let * denote convolution then

$$(2.4) \quad h(t) = f(t) * g(t) \Longrightarrow$$

$$H(\omega) = Odd[F(\omega)G(\omega)] + Even[F(\omega)G(-\omega)]$$

Here we have used the usual convention that uppercase variables refer to transformed functions. Note that if either F or G is even then the Hartley convolution property is the same as the familiar Fourier convolution property.

The above three properties of the continuous Hartley transform are all that we require to tackle the image reconstruction problem, given complete continuous projections. We will show how to go about it in the next section.

3. Reconstruction from continuous projections.

Following the terminology of Kak and Slaney [4], we use figure 1 below to define a *projection*. A straight line running through an image, f(x,y), is called a *ray*. The line integral of f(x,y) along this ray is called a *ray integral.* A set of ray integrals along parallel rays, all at angle θ to the x axis, forms a projection, $p_\theta(t)$.

The problem of image reconstruction is to reconstruct f(x,y) from a set of projections taken at various angles in the interval $[0,\pi]$. The projections are the input data and the reconstructed image, f(x,y) is the output. In practice we only have discrete data at our disposal, however in this section we will employ the continuous Hartley transform to reconstruct f(x,y) exactly from continuous projections.

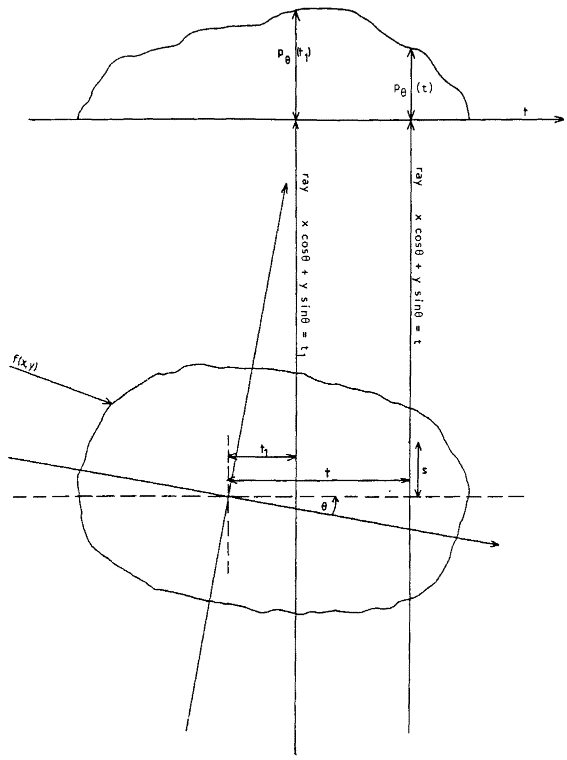

Figure 1: A projection through an image.

Suppose a projection through the image f(x,y) is measured at angle θ. The t^{th} ray of this projection is given by the equation

$$(3.1) \quad t = x \cos\theta + y \sin\theta$$

where t is the orthogonal distance from the ray to the origin. The projection at angle θ is a function of one variable, $p_\theta(t)$, defined by:

$$(3.2) \quad p_\theta(t) = \int_{ray_t} f(x,y)\ ds$$

$$= \int_{-\infty}^{\infty} f(t \cos\theta - s \sin\theta\ ,\ s \cos\theta + t \sin\theta) ds$$

There are several different approaches to the reconstruction problem and the interested reader is referred to a tutorial review by Mersereau and Oppenheim [11]. We limit our discussion to techniques based on the projection—slice theorem. This theorem was first derived using the Fourier transform and it relates the 1D transform of a projection to the 2D transform of the original image. We derive a Hartley transform equivalent of the projection—slice theorem.

Projection—slice theorem (Hartley transform)
The Hartley transform $P_\theta(w)$ of a projection $p_\theta(t)$ through an image f(x,y) is equal to the 2D Hartley transform F(u,v) of the image evaluated along the polar line $(u,v) = (\omega \cos\theta, \omega \sin\theta)$.

{ ie: for fixed θ, $F(\omega \cos\theta, \omega \sin\theta) = P_\theta(\omega)$ }

Proof
Let F(u,v) be the Hartley Transform of an image f(x,y), then the 2D Hartley inversion formula gives

$$(3.3) \qquad F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\,\mathrm{cas}(2\pi(ux+vy))\,dxdy$$

Now if $P_\theta(\omega)$ is the Hartley transform of $p_\theta(t)$ then using (3.2) we get

$$(3.4) \quad P_\theta(\omega) = \int_{-\infty}^{\infty} p_\theta(t)\ \mathrm{cas}(2\pi\omega t)\ dt$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t\cos\theta - s\sin\theta, s\cos\theta + t\sin\theta)\ \mathrm{cas}(2\pi\omega t)\,dsdt.$$

Now make the substitution

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix}$$

with unit jacobian to get:

$$(3.5) \qquad P_\theta(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\,\mathrm{cas}(2\pi(x\omega\cos\theta + y\omega\sin\theta))\,dxdy$$

$$= F(\omega \cos\theta\ ,\ \omega \sin\theta)$$

The projection—slice theorem in this raw form is seldom used for reconstruction algorithms since, when we only have discrete data available, equation (3.5) yields F on a polar grid and a 2D interpolation to a rectangular grid is required before Hartley inversion of F(u,v) can be attempted. However the theorem can be used to derive a reconstruction algorithm which only requires 1D interpolation to reconstruct from discrete data. We shall derive the Hartley equivalent of this *filtered back-projection algorithm.*

Consider the 2D inversion theorem for the Hartley transform:

$$(3.6) \qquad f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v)\ \mathrm{cas}(2\pi(ux+vy))\ dx\,dy\ .$$

Switching to polar coordinates, ie: $u = \omega \cos\theta$, $v = \omega \sin\theta$, jacobian $= |\omega|$, we get

$$(3.7) \qquad f(x,y) = \int_{0}^{\pi} \int_{-\infty}^{\infty} |\omega|\,F(\omega\cos\theta, \omega\sin\theta)\ \mathrm{cas}(2\pi\omega(x\cos\theta + y\sin\theta))\,d\omega d\theta$$

and making use of the projection—slice theorem (3.5) we get

$$(3.8) \qquad f(x,y) = \int_{0}^{\pi} \int_{-\infty}^{\infty} |\omega|\,P_\theta(\omega)\,\mathrm{cas}(2\pi\omega(x\cos\theta + y\sin\theta))\,d\omega d\theta$$

For notational convenience we rewrite equation (3.8) as

$$(3.9a) \quad f(x,y) = \int_{0}^{\pi} p_\theta^\omega(x \cos\theta + y \sin\theta)\ d\theta$$

where

$$(3.9b) \quad p_\theta^\omega(t) = \int_{-\infty}^{\infty} |\omega|\ P_\theta(\omega)\ \mathrm{cas}(2\pi\omega t)\ d\omega$$

These equations (3.9) describe the filtered back—projection algorithm in terms of the Hartley transform. Given projections, $p_\theta(t)$, we first compute *filtered* projections, $p_\theta^\omega(t)$, according to equation (3.9b) and then we smear each filtered projection back across the spacial plane at angle θ to build up f(x,y) according to (3.9a). Note that if $p_\theta^\omega(t)$ is only available at discrete points then 1D interpolation is all that is required to approximate $p_\theta^\omega(x \cos\theta + y \sin\theta)$. Equation (3.9b) together with the convolution theorem tells us that we can think of $p_\theta^\omega(t)$ as the convolution of $p_\theta(t)$ with the inverse Hartley transform of the function $|\omega|$. A similar result occurs in Fourier reconstruction theory [13] which is to be expected since $|\omega|$ is an even function and in this case the inverse Hartley transform is the same as the inverse Fourier transform. In practice the projection data consists of a finite number of projections each sampled at a finite number of points. In the next section we derive the Hartley equivalent of the sampling theorem which we need to adapt algorithm (3.9) for dealing with discrete data.

4. The sampling theorem for band limited functions.

A function g(t) will be called band–limited with bandwidth W if its Hartley transform $G(\omega)$ is zero whenever $|\omega| > W$. The sampling theorem states that if a band–limited function, g(t), is sampled at discrete points $\left\{ g(n\tau) \right\}_{n=-\infty}^{\infty}$ then g(t) may be recovered at intervening points according to

$$(4.1) \quad g(t) = \sum_{k=-\infty}^{\infty} g(k\tau) \frac{\sin 2\pi W(t-k\tau)}{2\pi W(t-k\tau)}$$

provided the sample spacing, $\tau = \frac{1}{2W}$.

Proof:
First we must establish a Hartley series expansion for $G(\omega)$ inside its support region. To this end consider the set of basis functions:

$$(4.2) \quad \varphi_k(\omega) = \begin{cases} \frac{1}{\sqrt{2W}} \operatorname{cas}(k\pi\omega/W) & \text{for } |\omega| < W \\ 0 \;\ldots\ldots\ldots\ldots & \text{for } |\omega| > W \end{cases}$$

This set is orthonormal in the sense that

$$\int_{-W}^{W} \varphi_k(\omega)\, \varphi_j(\omega)\, d\omega = \frac{1}{2W}$$

$$(4.3) \quad \int_{-W}^{W} \cos(\pi\omega(k-j)/W)\, d\omega = \delta_{kj}$$

which allows an expansion for $G(\omega)$ inside its interval of support as

$$(4.4a) \quad G(\omega) = \sum_{k=-\infty}^{\infty} G_k\, \varphi_k(\omega)$$

where

$$(4.4b) \quad G_k = \int_{-W}^{W} G(\omega)\, \varphi_k(\omega)\, d\omega.$$

Now to get the sampling theorem we substitute (4.2) into (4.4a) into Hartley's inversion formula (2.3), to get:

$$(4.5) \quad g(t) = \int_{-W}^{W} G(\omega) \operatorname{cas}(2\pi\omega t)\, d\omega$$

$$= \sum_{k=-\infty}^{\infty} \sqrt{2W}\, G_k \frac{\sin(2\pi W(t-k\tau)}{2\pi W(t-k\tau)}$$

provided $t \neq n\tau$ for some integer n. When $t = n\tau$ the same substitutions result in:

$$(4.6) \quad g(n\tau) = \sqrt{2W}\, G_n$$

which allows us to rewrite (4.5) as

$$(4.7) \quad g(t) = \sum_{k=-\infty}^{\infty} g(n\tau) \frac{\sin(2\pi W(t-k\tau)}{2\pi W(t-k\tau)}$$

for all t, and thus the sampling theorem is established in a Hartley setting. In the next section we will make use of this sampling theorem to discretize the reconstruction algorithm of equations (3.9).

5. Discretizing the filter.

In order to discretize equation (3.9a) one must assume that the projections are band–limited. So, assuming that $P_\theta(\omega) = 0$ when $|\omega| > W$

we make use of the sampling theorem to express $P_\theta(\omega)$ as

$$(5.1)\quad P_\theta(\omega) = \int_{-\infty}^{\infty} p_\theta(t') \; \mathrm{cas}(2\pi\omega t') \; dt' =$$

$$\int_{-\infty}^{\infty} \left\{ \sum_{k=-\infty}^{\infty} p_\theta(k\tau) \frac{\sin\; 2\pi W(t'-k\tau)}{2\pi W(t'-k\tau)} \right\} \mathrm{cas} 2\pi\omega t' dt'$$

Substitution of this expression in equation (3.9b) yields

$$(5.2)\qquad p_\theta^\omega(t) =$$

$$\tau \sum_{k=-\infty}^{\infty} p_\theta(k\tau) \int_{-W}^{W} |\omega| \; \cos\; 2\pi\omega(k\tau - t) \; d\omega$$

after much manipulation.

To obtain equation (5.2) one needs the following Hartley transform of the sinc function:

$$(5.3)\quad \int_{-W}^{W} \frac{\sin\; 2\pi W(t'-k\tau)}{2\pi W(t'-k\tau)} \; \mathrm{cas}(2\pi\omega t') \; dt'$$

$$= \begin{cases} \tau \; \mathrm{cas}(2\pi\omega k\tau) & \text{when } |\omega| < W \\ 0 \; \ldots\ldots\ldots & \text{when } |\omega| > W \end{cases}$$

This result is quite tedious to prove and will be left to the reader.

Continuing with the discretization process, to obtain $p_\theta^\omega(t)$ at N sample points, $\left\{ p_\theta^\omega(n\tau) \right\}_{n=0}^{N-1}$, we let $t = n\tau$ in equation (5.2) to get:

$$(5.4)\qquad p_\theta^\omega(n\tau) =$$

$$\frac{1}{4\tau} p_\theta(n\tau) \; - \tau \sum_{\substack{k=-\infty \\ k\neq n \text{ and } k \text{ odd}}}^{\infty} \frac{p_\theta(k\tau)}{\tau^2\pi^2(k-n)^2}$$

which we can write as the discrete convolution:

$$(5.5)\quad p_\theta^\omega(n\tau) \;=\; \tau \sum_{k=0}^{N-1} q(n\tau - k\tau) \; p_\theta(k\tau)$$

$$\text{where}\quad q(n\tau) = \begin{cases} 1/4\tau^2 & ;\quad n = 0 \\ 0 & ;\quad n \text{ even}. \\ -1/n^2\pi^2\tau^2 & ;\quad n \text{ odd} \end{cases}$$

provided $p_\theta(k\tau)$ is zero when $k < 0$ or $k \geq N$. Exactly the same expression is obtained by Kak and Slaney [4] using the Fourier sampling theorem. In section 7 we will show how to use the discrete Hartley transform to compute the discrete convolution required by equation (5.5). However, for the sake of completeness, in section 6 we first show how to discretize the back–projection of equation (3.9a).

6. Discretizing the back–projection.

For this part of the algorithm we followed Kak and Slaney [4] exactly. Discrete approximation of the integral (3.9a) yields .

$$(6.1)\quad f(x,y) = \frac{\pi}{N} \sum_{i=1}^{N} p_\theta^\omega(\; x \cos\theta_i + y \sin\theta_i \;)$$

where the angles θ_i are those for which the projections $p_\theta(n\tau)$ have been measured. Since the value of $x \cos\theta_i + y \sin\theta_i$ in equation (6.1) may not correspond to one of the values of $n\tau$ for which $p_\theta^\omega(n\tau)$ is determined by equation (5.5), we make use of linear interpolation to estimate $p_\theta^\omega(x \cos\theta_i + y \sin\theta_i)$.

7. Computing a convolution using the discrete Hartley transform

Bracewell in [1] introduced the discrete Hartley transform specifically for dealing with convolutions such as the one in equation (5.5). Bracewell's discrete Hartley transform for a discrete real function, $h(k\tau)$, is given by:

$$(7.1)\quad H(j\Gamma) \;=\; \frac{1}{N} \sum_{k=0}^{N-1} h(k\tau) \;\; \mathrm{cas}(2\pi kj/N)$$

where $\Gamma = \dfrac{1}{N\tau}$.

Simple substitution gives the discrete inverse formula as

$$(7.2) \quad h(k\tau) = \sum_{j=0}^{N-1} H(j\Gamma)\ \text{cas}(2\pi jk/N).$$

Bracewell goes on to show that discrete convolutions such as the one in equation (5.5) may be calculated efficiently by employing the following discrete convolution theorem:

Discrete convolution theorem:

If $\left\{\begin{array}{c} h_1(k\tau) \\ \Updownarrow \\ H_1(j\Gamma) \end{array}\right\}$ and $\left\{\begin{array}{c} h_2(k\tau) \\ \Updownarrow \\ H_2(j\Gamma) \end{array}\right\}$ are

discrete Hartley transform pairs then so to is:

$$\begin{array}{c} h_1(k\tau) \otimes h_2(k\tau) \\ \Updownarrow \\ \frac{N}{2}\Big[H_1(j\Gamma)H_2(j\Gamma) - H_1(N-j\Gamma)H_2(N-j\Gamma) + \\ H_1(j\Gamma)H_2(N-j\Gamma) + H_1(N-j\Gamma)H_2(j\Gamma)\Big] \end{array}$$

where the symbol $\otimes$ denotes circular convolution.

The above convolution theorem simplifies considerably if one of the sequences involved, $h_2(k\tau)$ say, is even in the sense that $h_2(N-k\tau) = h_2(k\tau)$. If this is the case it is a simple matter to show that its discrete Hartley transform, $H_2(j\Gamma)$, is even in the same sense and the convolution theorem reduces to:

Discrete convolution theorem for symetrical sequences:

If $\left\{\begin{array}{c} h_1(k\tau) \\ \Updownarrow \\ H_1(j\Gamma) \end{array}\right\}$ and $\left\{\begin{array}{c} h_2(k\tau) \\ \Updownarrow \\ H_2(j\Gamma) \end{array}\right\}$ are

discrete Hartley transform pairs then so to is:

$$\left\{\begin{array}{c} h_1(k\tau) \otimes h_2(k\tau) \\ \Updownarrow \\ N\Big[\ H_1(j\Gamma)\ H_2(j\Gamma)\ \Big] \end{array}\right\}$$

This theorem indicates that in order to perform a discrete circular convolution of two sequences one of which exhibits even symmetry, first compute the discrete Hartley transform of each sequence, then multiply the resulting sequences together, then compute the inverse discrete Hartley transform. The convolution required in equation (5.5) is not circular but this drawback can be overcome by the usual method of zero—padding. Indeed, judicious placing of the padding zeros will ensure that the sequence $q(n\tau)$ retains even symmetry allowing the simplified theorem to be used in the convolution computation. The algorithm for computing the filter part of the reconstruction algorithm can now be written as:

$$(7.3) \quad p_\theta^\omega(n\tau) = \tau \times N \times \text{DHT}\Big\{\text{DHT}\{p_\theta(n\tau)+\text{ZP}\} \times \text{DHT}\{q(n\tau)+\text{ZP}\}\Big\}$$

where DHT indicates discrete Hartley transform, ZP indicates zero—padding, and N is the number of data points in $p_\theta(n\tau)$ plus the number of zero paddings.

8. Testing procedures

To test our Hartley reconstruction procedure we used the so—called Shepp—Logan head phantom [6] as a test image. This image, shown below in the top left position of plate 1, has been used since 1974 to test many reconstruction procedures. The test image is built up by superimposing ellipses of different attenuation. Kak and Slaney [4] give an algebraic formula for calculating the projection through any ellipse. Using this formula and the fact that the projection operator is linear we can compute exactly any projection through the test image. We thus have a complete set of data on which to test our reconstruction procedure.

In our reconstruction experiments we first calculated $2^{\gamma-1}$ projections sampled at $2^{\gamma-1}-1$ evenly spaced sample points. Reconstruction then consists of padding with

$2^{\gamma-1}+1$ zeros, applying the filter via the discrete transforms given in equation (7.3) and finally performing the back–projection quadrature as outlined in section 6. The transforms in (7.3) involve 2^{γ} points and are accomplished using Bracewell's fast Hartley transform algorithm published in [1]. In plate 1 reconstructions are shown for $\gamma = 7$ (top right), $\gamma = 8$ (bottom left) and $\gamma = 9$ (bottom right).

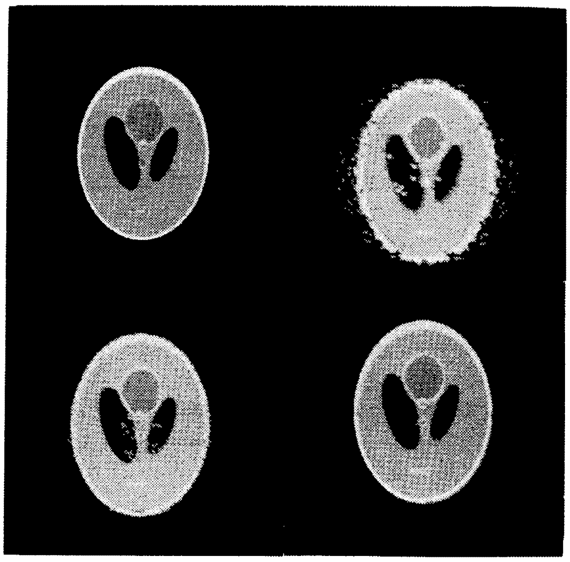

Plate 1: Head phantom reconstructions

Top left : Shepp Logan test image
Top right: Reconstruction, $\gamma = 7$.
Bot left : Reconstruction, $\gamma = 8$.
Bot right: Reconstruction, $\gamma = 9$.

In the table below we compare the filtered back–projection reconstruction algorithm based on Bracewell's FHT published in [1] with the standard filtered back–projection algorithm based on Borland's Turbo Pascal FFT from Borland's numerical toolbox, [2].

Gamma	FFT	FHT
7	36	33
8	95	80
9	286	221

Table 1: Reconstruction times in seconds:

These timings were measured on an HP 9000 runing HP Pascal under UNIX. A 22% improvement is obtained for a 256 by 255 reconstruction ($\gamma = 9$).

Section 9: Conclusion

The FHT offers a faster method for performing convolutions of real sequences than the standard FFT offers. The Hartley transform has dual theorems for all Fourier transform theorems. These two facts indicate that any algorithm based on FFT convolution of real data can be improved by employing the FHT in place of the FFT. We have demonstrated that filtered back–projection reconstruction can be improved in this way.

References:

[1] R.N. Bracewell, The Hartley Transform, Oxford University Press, 1986.

[2] Borland, Turbo Pascal Numerical Methods Toolbox, Borland Press, 1986

[3] R.V.L. Hartley, A more symmetrical Fourier Analysis applied to transmission problems, Proc. Inst. Radio Engrs., March 1942, pp. 144–150.

[4] A.C. Kak and M Slaney, Principles of Computerized Tomographic Imaging, IEEE Press,1987.

[5] H.J. Meckelburg and D Lipka, Fast Hartley Transform Algorithm, Electronics Letters, Vol. 21, No. 8, April 1985, pp. 341–343.

[6] L.A. Shepp and B.F. Logan, The Fourier Reconstruction of a Head Section, IEEE Trans. Nucl. Sc., Vol NS–21, June 1974, pp. 21–43.

Computers and the Law

Submitted by Antony Cooper
CSIR

The SA Law Commission has established a commission on "The Legal Protection of Information".

The commission is still in its preliminary stages and the assigned researcher, Mr Herman Smuts, is still preparing the working paper. He does not know when it will be finished, but once the working paper has been prepared, they will invite comments for about two years, before preparing the final report. I have contacted Mr Smuts, and he would be most grateful to receive input at this stage, especially regarding the terms of reference of the commission. His address is:

C/o SA Law Commission
Private Bag X668
PRETORIA
0001

In addition, there is an ad-hoc committee at the Registrar of Copyright investigating numerous copyright issues, including those relating to software and data. Mr Smuts' commission will be liaising with the ad-hoc committee.

I feel that SAICS has an obligation to submit evidence to the commission, and I would appreciate it if you would circulate the members of the Council of SAICS, and perhaps the general membership as well, to solicit ideas concerning SAICS's input.

I shall prepare something for the commission, either in my personal capacity, or in my professional capacity here at CSIR. I would be willing to assist in the preparation of any evidence SAICS might submit.

4th National MSc/Phd Computer Science Conference

Report by Danie Behr
University of Pretoria

This conference was held from 7th to 10th September 1989 at the Cathedral Peak Hotel in the Drakensberg. The conference was attended by 61 postgraduate students from 11 South African universities. Most were engaged in MSc studies, although 5 Phd students also attended. These numbers are encouraging for the South African computer science community. This type of conference is rather unique in that it affords students the opportunity of sharing their research, and getting to know other researchers in the country. The number of Afrikaans and English speaking students attending the conference were roughly equal. Presentations were made in the language preferred by the student. Invitations were sent to all universities with computer science departments. The conference was organized by the students themselves.

Some of the more popular research topics that were presented included expert systems, data communications, computer security, graphics, software engineering, user interfaces and data bases. The main sponsor for this year's conference was the Division for Microelectronic Systems and Communication Technology of the CSIR. The conference was opened with an interesting talk on the myths and motivations of post graduate studies by Prof DG Kourie, acting head of the Computer Science Department at Pretoria University.

The next conference will be presented by the University of Port Elizabeth. People requiring further information about the next conference should contact Andre Calitz, Charmaine du Plessis or Jean Greyling of the Department of Computer Science at UPE.

A list of authors and papers presented at the symposium follows:

S Crosby, University of Stellenbosch
Performance Analysis of Wide Area Computer Communication Networks

A B Joubert, PU for CHE Vaal Triangle Campus
Image Processing Libraries

A Calitz, University of Port Elizabeth
An Expert System Toolbox to assist in the classification of objects

L von Backström, University of Pretoria
Integrated Network Management

R Foss, Rhodes University
The Rhodes Computer Music Network

A McGee, University of Natal
On Fixpoints and Nondeterminism in the Sigma-Lambda Calculus

P G Mulder, Randse Afrikaanse Universiteit
A Formal Language and Automota approach to Data Communications

A Tew, Randse Afrikaanse Universiteit
Drie dimensionele grafiek grammatikas

T C Parker-Nance, University of Port Elizabeth
Human-Computer Interaction: What Determines Computer Acceptance

E Coetzee, PU vir CHO Vaaldriehoekkampus
Opsporing van rande in syferbeelde dmv verskerping en drempelbepaling
D A Sewry, Rhodes University
Visual Programming
A Cooper, University of Pretoria
Improvements to the National Exchange Standard
E S Badier, University of Port Elizabeth
A Computer Assisted Diagnostic System (CADS)
C du Plessis, Universiteit van Port Elizabeth
Persoonsidentifikasie dmv naampassing in 'n genealogiese databasis
J Greeff, University of Stellenbosch
The Entity-Relationship Model and its Implementation
D A de Waal, PU vir CHO
Flat Concurrent Prolog (FCP) en Flat Guarded Horn Clauses (FGHC): 'n Vergelyking
E Naude, UNISA
Interne metodes in Liniêre Programmering
A Deacon, University of Stellenbosch
Global consistency in non-locking DDBMS
A Wilks, Rhodes University
The Synchronisation and Remote Configuration of the Resources in a Computer Music Network
J Greyling, University of Port Elizabeth
The design of a User Interface with special reference to an Interactive Molecular Modelling Program
L Drevin, PU vir CHO
Rekenaarsekuriteit: Verskillende vlakke van kontrole
Dieter C Barnard, University of Stellenbosch
The design and implementation of a modest, interactive proof checker
R A Schmidt, University of Cape Town
Knowledge Representation Systems and the Algebra of Relations
J Hartman, Randse Afrikaanse Universiteit
Die Gebruik van Objek-georiënteerde Programmering in die Moderne Sneltrein Omgewing
S Lawrie, Rhodes University
The Design and Implementation of a System for the Interactive Control of a MIDI-based Studio
E Mulder, Rand Afrikaans University
A Formalisation of Object-Oriented Principles
C J Tolmie, UOVS
Die Ontwikkeling van 'n Ekspertrekenaarstelsel vir die beoordeling van die resultate van die Technicon H1-Bloedselanaliseerder
R Breedt, University of Pretoria
Realism with Ray Tracing
J van Jaarsveld, University of Pretoria
Developing Medical Expert Systems: A knowledge acquisition perspective
W Appel, University of Pretoria
TCP/IP Implementation on Ethernet
E Goedeke, University of Natal
Eggspert's Control Structure
M Harmse, University of Stellenbosch
Modelling of I/O Subsystems
H L Viktor, University of Stellenbosch
A Quantitative Model for Comparing Recovery Techniques in a Distributed Database
M Olivier, Randse Afrikaanse Universiteit
Rekenaarvirusse in Suid-Afrika

Book Reviews

An Introduction to Functional Programming Through Lambda Calculus
by Greg Michaelson, Addison-Wesley, 1988.
Reviewer: Dr. E P Wentworth, Rhodes University

Recently we have seen a number of excellent *second generation* texts on Functional Programming. Michaelson's text assumes some previous programming experience with imperative languages, and presents the functional approach as an alternative paradigm. He begins with a very accessible exposition of the Lambda Calculus, and carefully develops this foundation to encompass the important aspects and paradigms of functional programming. The programming notation is language-independent, although the last chapters are devoted to a brief look at two specific languages, Standard ML and Lisp. The examples and exercises are mainly utility in nature, e.g. "insert a sublist after the first occurrence of another sublist in a list", and can generally be solved in a couple of lines. Answers to the exercises are provided in an appendix.

The approach is slanted towards developing a solid base for understanding functional languages and computing. In this respect the book achieves a good balance between the theoretical underpinnings and their practical application. On the practical side, however, I found the lack of more substantial examples and exercises disappointing. Most programming texts tackle a set of 'standard' problems which are well-understood in the academic community and provide an informal benchmark for comparisons. Since the book is targeted for those already versed in imperative languages and standard algorithms, one might expect the examples to clearly demonstrate the elegance and power of the *problem-oriented* functional approach in these areas. Having laid an excellent foundation I was left with the feeling that the book failed to capitalize and deliver the cherry on the top.

The book is highly recommended as one of the new breed of Computer Science books which gives substantial attention to the fundamentals of the subject without becoming bogged down in over-rigorous formality.

Artificial Intelligence and the Design of Expert Systems
by George F Luger & William A Stubblefield, The Benjamin/Cummings Publishing Co., 1989.
Artificial Intelligence: A Knowledge-based Approach
by Morris W Firebaugh, PWS-Kent Publishing Co., 1989.
Reviewer: Prof G D Oosthuizen, University of Pretoria

One of the primary goals of an Honours course is to introduce students to a field in such a way that they arrive at enough insight into relevant issues to enable them to conduct further research on their own. To this end a text book which is used ought to reflect the current view of the field. Because of the rapid expansion of the field of Artificial Intelligence (AI), we have now finally outgrown the era dominated by the books by Winston and Charniak and McDermott. In the past five to ten years much new work has been done, and new insights have been gained. Introducing AI, therefore, requires a marked shift from the previous emphasis on a few historical systems embodying a number of famous methods, to a more generic approach - an approach which highlights those fundamental representation and search models that span all the different application areas and strategies of problem solving. Of course, since AI still does not have a well developed theory, references to seminal systems continues to fulfil an important role.

Both of the above books are good text books, characterised by a balanced coverage of Prolog and Lisp. They also reflect and consolidate much of the work of the past few years done in areas such as knowledge representation, machine learning, the work done under the heading of Expert Systems and even the recent work on neural networks. But the most important feature that they share is the accurate and up to date overall picture of the subject provided; the broad framework for the understanding of AI that is created without neglecting work of historical importance. There are still references to these works, but they are placed in perspective in relation to new developments.

The book of Luger & Stubblefield (L&S) is more language oriented than Firebaugh's book. A characteristic of L&S is that AI approaches to representation are related to the Object Oriented approach. Whereas L&S includes chapters on advanced AI programming techniques in Prolog and Lisp, it does not address pattern recognition, computer vision and robotics. (Firebaugh has chapters on each of these themes.) These omissions are understandable, since AI has diversified so much recently that it is difficult to cover all applications in one book.

If I had to select one of the books, it would be L&S. Although L&S gives poor coverage of Machine Learning, the book's overall presentation is very good. In particular, the chapters are well-organised, and the overall approach to AI - starting with the core aspects of *representation* and *search*, followed by chapters on AI languages - is coherent. The authors also make very good use of graphical representations and illustrations to convey ideas.

Books Received

The following books have been sent to SACJ. Anyone willing to review a book should contact the editor. The book will be sent to him for review, and may be kept provided that a review is received.

- D Bustard, J Elder & J Welsh, [1988], *Concurrent Program Structures*, Prentice-Hall Inc., Englewood Cliffs.
- R Cafolla & A D Kauffman, [1988], *Turbo Prolog Step by Step*, Merrill Publishing Company, Columbus, Ohio.
- S Hekmatpour, [1988], *Introduction to LISP and Symbol Manipulation*, Prentice-Hall Inc., Englewood Cliffs.
- K L Clark & F G McCabe, [1984], *micro-PROLOG: Programming in Logic*, Prentice-Hall Inc., Englewood Cliffs.
- D Crookes, [1988], *Introduction to Programming in Prolog*, Prentice-Hall Inc., Englewood Cliffs.
- M J C Gordon, [1988], *Programming Language Theory and its Implementation*, Prentice-Hall Inc., Englewood Cliffs.
- J G Hughes, [1988], *Database Technology : A software engineering approach*, Prentice-Hall Inc., Englewood Cliffs.
- R Milner, [1989], *Communication and Concurrency*, Prentice-Hall Inc., Englewood Cliffs.
- T J Myers, [1988], *Equations, Models and Programs*, Prentice-Hall, Inc., Englewood Cliffs.
- N C Rowe, [1988], *Artificial Intelligence through Prolog*, Prentice-Hall Inc., Englewood Cliffs.
- D A Protopapas, [1988], *Microcomputer Hardware Design*, Prentice-Hall Inc., Englewood Cliffs.
- H Eisner, [1988], *Computer-aided Systems Engineering*, Prentice-Hall Inc., Englewood Cliffs.
- S H Unger, [1989], *The essence of logic circuits*, Prentice-Hall Inc., Englewood Cliffs.
- R J Young, [1989], *Practical Prolog*, Van Nostrand Reinhold, New York.

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as a Communications or 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 double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - 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 on separate sheets of A4 paper, and should be numbered and titled. Figures should be submitted as original line drawings, and not photocopies.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin. Distinguish clearly between such cases as:
  - upper and lower case letters;
  - the letter O and zero;
  - the letter I and the number one; and
  - the letter K and kappa.
- 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 in square brackets. References should thus take the following form:

  [1] E Ashcroft and Z Manna, [1972], The translation of 'GOTO' programs to 'WHILE' programs, *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.

  [2] C Bohm and G Jacopini, [1966], Flow diagrams, Turing machines and languages with only two formation rules, *Comm. ACM*, **9**, 366-371.

  [3] S Ginsburg, [1966], *Mathematical theory of context free languages,* McGraw Hill, New York.

Manuscripts *accepted* for publication should comply with the above guidelines, and may provided in one of the following three formats:

- in a **typed form** (i.e. suitable for scanning);
- as an **ASCII file** on diskette; or
- **in camera-ready** format.

A page specification is available on request from the editor, for authors wishing to provide camera-ready copies.

Charges

A charge per final page, scaled to reflect scanning, typesetting and reproduction costs, will be levied on papers accepted for publication. The costs per final page are as follows:

Typed format:	R80-00
ASCII format:	R60-00
Camera-ready format :	R20-00

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

Proofs

Proofs of accepted papers will 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.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

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 specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

South African Computer Journal
Number 1, January 1990
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif
Nommer 1, Januarie 1990
ISSN 1015-7999

Contents