Issue I September 1 2002

The Ring

The Newsletter of the University of Cambridge Computer Laboratory Graduate Association

William Gates Building Opens

By Professor Ian Leslie

The 1st of May saw the successful culmination of over four years of planning and building as the new Computer Laboratory officially opened its doors.

There was a large and enthusiastic turnout as around 400 staff, students - past and present - and friends congregated to mark the launch of the new state-of-the-art building which will not only support the continued innovation in computer science and groundbreaking research for which Cambridge University is known, but will also reinforce Cambridge as the European capital of high technology.

It was fitting that Professor Sir Maurice Wilkes performed the opening ceremony by unveiling the old lab's original front door. Sir Maurice, who while Head of the Laboratory developed the famous EDSAC, said "This door, standing in this new laboratory, represents the door to the future". This sentiment was echoed both by the event's host, the Vice-Chancellor Sir Alec Broers and by the then Financial Secretary to the Treasury, the Rt Hon Paul Boateng MP who we were delighted could attend.

I would again like to express my appreciation and gratitude to the wide group of people involved with the project, not least to those whose generosity made it all possible.

If you were not able to join us in May, we would be delighted to welcome you to the new Laboratory and hope you will come to the many future events we shall be holding.

Welcome to the Cambridge Computer Lab Ring

The Cambridge Computer Lab Ring (www.camring.ucam.org) – the Computer Laboratory's graduate association named after the 10 megabit Cambridge Ring, which came into use in 1975 and was among the first local area networks to be commercially available - provides the means to link life as a student to the lives we experience as professionals. Not only does it offer the opportunity to network with old classmates but it also provides opportunities to meet and network with other entrepreneurs and industry professionals. Members can establish relationships that can give them guidance, support, connections to venture capital investors and other professional organizations. Monthly events not only represent part of a linked network that can be rewarding for each participant but they can also help to ensure that the prestigious status of the Computer Laboratory is further enhanced.

So join now by contacting Jan Samols on 01223 763 585 or emailing Jan.Samols@cl.cam.ac.uk and providing your full name, course, year of graduation and membership category required (see below).Alternatively you can send your details to the Cambridge Computer Lab Ring Office, William Gates Building, JJ Thomson Avenue, Cambridge, CB3 0FD.
The benefits and services of membership include:

Online Directory

The directory gives members password protected online access to the Cambridge in Computing network: a self-maintained membership which facilitates contact with other members.

Newsletter

All members will receive a newsletter 3 times a year, enabling them to keep abreast of news of the Laboratory, association events and profiles of their fellow graduates. Make sure your news is included in the next edition by sending it to the Cambridge Computer Lab Ring (see above for contact details)

Invitations to the social, technical and business events Programme

Our first event will be a professional seminar on

October 10 2002. Further details can be found overleaf.

Access to the Laboratory's weekly seminar programme

For further details contact Anuj Dawar at Anuj.Dawar@cl.cam.ac.uk or phone 01223 331786.

Access to the Laboratory's specialist library

If you are not able to visit the library yourself, the library service may be able to photocopy and send you the information you are looking for.

Career advice

Use the online directory to network with Laboratory graduates. If you're looking for advice on the telecoms industry or starting up a new company, there's someone on the network who can help.

If you are willing to offer informal careers advice to students and graduates please contact us (see above for details).

If you are working and your organisation requires students for work placements or has graduate vacancies, you will be able to submit a job posting on the associations's website.

Mentoring programme

We hope to be able to offer a student/graduate mentoring programme whereby students of the Laboratory are matched with graduate mentors.

If you would like to take part in the Mentoring Programme and would like further information, please contact us (see above for contact details)

Helpline

Whether you seek information on careers, work permits or starting your own company, the Cambridge Computer Lab Ring office is open to answer your calls.

Membership category

Current students and those who have graduated within the last 3 years – FREE
Single UK Membership £48 (£35 if resident in Scotland or N.Ireland)
Overseas Membership £25
Senior Membership (over 65) £35
Associate Membership[1] £75
Life Membership £800; Senior Life membership £300; Overseas Life Membership £400; Associate Life Membership £1200

[1] Any Cambridge graduate who is now working within the computer industry or within a computer-related department of any business or institution

Cambridge Computer Lab Ring Launch Event

The opening event of the Cambridge Computer Lab Ring will take place on October 10 at 16:15 in Lecture Theatre 2 of the William Gates Building.

The seminar will be hosted by Stephen Allott (the President of the Cambridge Computer Lab Ring and the Laboratory's Director of Development) and will feature a talk by Laurence Garrett of 3i entitled "The Cambridge Technology Scene – past, present, and future".

Laurence Garrett joined 3i plc in 1994 on his return from the San Francisco office of Deloitte & Touche. With a degree in Computer Science, Laurence immediately specialized in technology making the investments that led to the IPOs on NASDAQ and the LSE of Select Software Tools Inc, Weston Medical plc, and Marlborough Stirling plc. He is currently the local director for the Cambridge office of 3i, managing 6 investors and directly looking after 3i's investments in Cambridge Sililcon Radio and Cambridge Positioning Systems.

The talk will cover some history on the Cambridge technology market, the successes that have led to the title "Silicon Fen", a snapshot of the current market and finally a view on what venture capitalists are looking for in order to seed the next generation of successful tech companies in the region.

We hope that you will be able to join us on the 10th. If you require further information, please contact the Cambridge Computer Lab Ring office on 01223 763585 or email Jan.Samols@cl.cam.ac.uk.

Student Awards

by Dr Simon Moore, Chief Examiner 2002

The Computer Laboratory continues to attract very bright undergraduates to study for the Tripos and Diploma in Computer Science. This is reflected in the continued support from the BCS and IEE who continue to accredit both the Tripos and the Diploma. We were also gratified that one of our Part II students last year – Hanna Wallach – won the Best Computer or Computer Software Student award in the national Science, Engineering & Technology Student of the Year competition for her final year project on "Diagrammatic integration of abstract operations into software work contexts".

From this year's Part II class, we have nominated the following students for the Science, Engineering & Technology Student of the Year competition due to the high quality of their final year projects:

Andrew Clark (King's College)
Optimising the Web for a GPRS Link

John N Foster (Emmanuel College)
Model Checking for a Functional Hardware Description Language

Barnaby G R Gray (Churchill College)
A Distributed Caching System

David Hart (Robinson College)
Portable Computer Based on a Custom Processor

Timothy M Hospedales (Jesus College)
Eye Tracker

David Spence (St John's College)
BCPL to C Converter

The range of subject areas covered by these projects reflects the broad scope of our teaching activities which in turn is backed up by leading research.

Prize winners

AT&T Prize for the top first in the Part II examinations
David Spence (St John's College)

Industiral Supporters Prize for the top first in the Part IB examination
Andrew Naish-Guzman (Gonville and Caius College)

The Data Connection Prizes for achievement in Part IA were awarded to:

D T Morgan (Gonville and Caius College)
P J Tuddenham (Gonville and Caius College)
V Vefeiadis (Selwyn College)

The AT&T Prize for top Diploma Student
Gareth Taylor (Corpus Christi College)

The Data Connection Prizes for the best dissertation in the Diploma Course were awarded to:

Xavier Lubino (Churchill College): "Crowbar: a Protocol for Opening Holes in Firewalls"

David Negrier (Churchill College): "Finding an Algorithm for Splitting Terrain Marshes"

Eiko Yoneki (Lucy Cavendish College): "MobileGateway with Publish/Subscribe Paradigm over Wireless Network"

PhD Update

By Dr Peter Robinson, PhD Coordinator

The Computer Laboratory now has an establishment of 30 academic staff and has a further 23 research workers on the payroll, but has 75 post-graduate research students working towards the PhD degree. These students continue to be the powerhouse of research in the Laboratory.

They also bring honour to the Laboratory – five of our graduates have won BCS Distinguished Dissertation Awards, out of only 30 awards in the 11 years that the scheme has run.

Unlike many other universities, we continue to be approached by large numbers of excellent students who would like to work in the Laboratory. We received 168 applications for October 2002 and have made offers to 53 of these. However, funding for research students is becoming an ever more serious problem.

The EPSRC allowed us just three studentships this year and the University gave us a further Domestic Research Studentship. The Laboratory's Local Industry Supporters have been very generous in their donations which will fund a further three students from overseas. Several other students have won independent studentships and some will take out loans, but we are likely to lose half the students that we would have liked to take.

Just over half of our research students now come from outside the UK and our international guests bring particular luster to the Laboratory. We are lucky to be able to support them through the Neil Wiseman Fund. This was established in 1995 to commemorate Neil and has received donations of over £100k in subsequent years. Unfortunately, this fund is now running out.

The full cost of a research student for three years from October 2002 is only £47k for a home or European Union student, or £69k for an international student from outside the EU. Any ideas for ways to raise this money (or even cash contributions!) would be most welcome.

MPhil Update

By Ann Copestake

The MPhil in Computer Speech and Language processing was replaced in 2001 with a Masters course in Computer Speech Text and Internet Technology (CSTIT) (http://svr-www.eng.cam.ac.uk/cstit/). Like the earlier course, it provides training in the theory of speech and language technology, but now also includes special emphasis on the use of the techniques in advanced Internet applications. The new course can be taken on either a full-time or part-time basis: part-time students come to Cambridge for one and a half days a week and complete the course in two or three years, as opposed to one year for full-time students. The course is jointly run by the Speech Vision and Robotics (SVR) group in the Department of Engineering and the Natural Language and Information Processing (NLIP) group in the Computer Laboratory. The course has its own Industrial Supporters Group consisting of companies with active interests in speech and language technology: for further information about the group, please contact cstit-enquiries@eng.cam.ac.uk.

The first year of the new course has been extremely successful. There were around 160 very high-quality applications for the 23 places (21 full-time, 2 part-time). The first two terms of the full-time course involve lectures and practicals, followed by a research project which is submitted at the end of July. This year's projects included a multi-modal tourist information application, a system for distinguishing between objective and subjective articles in newspapers and a system which morphs spoken voice into song. There were also several projects which involved more basic research, including what is believed to be the first algorithm for fully automatic discovery of word senses from large amounts of general written text. After submitting their theses, the students presented their projects on July 30th and 31st: the presentations were attended by industrial participants and other researchers as well as MPhil staff and students. It was generally agreed that the projects were of exceptionally high quality.

We have had even more applications for 2002-2003 and believe we have again attracted very high-quality students, though more applicants for part-time places would have been welcome. The administration of the course will move to the Computer Laboratory in September, so the URL given above

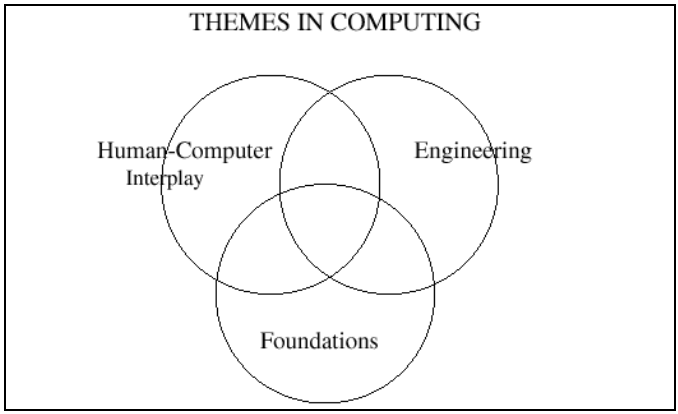
will soon be replaced with a link to a Lab web page. We are aiming to make the course accessible for distance learning and a Web development officer is being recruited to help in this process. All lectures and practicals will now take place in the Gates Building, with the practicals using workstations kindly donated by Hewlett-Packard.

Computing in Space

A lecture by Robin Milner, on the occasion of the opening of the Williams Gates Building for the University of Cambridge Computer Laboratory on 1 May 2002

Introduction

When I joined the Computer Laboratory in '95, and especially when I took over as Head in '96, I tried to comprehend what goes on here. This was a rewarding task; I learned a huge amount just by getting a grasp of how all things we do in the Lab relate to each other. In a very broad sense, one can classify the whole intellectual endeavour of Computer Science into three large overlapping themes: Engineering, Human-Computer Interplay, and Foundations.



Within Engineering, you have not only hardware but software engineering, and also communications; within Human-Computer Interplay you have not only linguistic and graphical interfaces, but also artificial intelligence; within Foundations you have not only the structure and complexity of algorithms, but also the semantics underlying computational processes and the logical tools to verify them.

It's a tribute to all my colleagues, and especially my immediate predecessor as Head, Roger Needham, that one could find ten or a dozen substantial activities in the Lab - all successful - spanning this space in an extraordinarily balanced way, and informing each other. At that time I drew a diagram of these research groups and their interaction on my blackboard, for a visit by Geoff Hoon, then a science minister (now Secretary of State for Defence). I think it was effective, and the same picture works just as well

with incoming PhD students. The story is much the same today, mutatis mutandis; we can be intensely proud of our broad and equal span of the subject.

On an occasion like this, we ought to wonder what we might achieve in, say, the next two decades, say. I'm not qualified - perhaps none of us is - to predict across such a wide spectrum of activity. But I shall stick my neck out, which is rare for a theoretician, and speculate on one way in which we might change the face of our subject in response to modern technology, rather than simply adapt to it. More forcefully, in this lecture I put forward a Grand Challenge which Computer Scientists might address, and this Lab is well placed to address it.

COMPUTING IN SPACE – EXAMPLES

- Multimedia Infrastructure for a Building
- Mobile Telephone Systems
- Molecular Computers in Biology

The Building

Before we moved into this new building we had long discussions about how we could link up all the lecture halls, teaching rooms and meeting rooms, using advanced audiovisual technology, to give a rich and flexible platform for teaching and research - and at the same time provide a live laboratory for our own research into computational platforms for multimedia. My colleague Peter Robinson had, and no doubt still has, a strategy for this. We had hoped for financial support from the Joint Infrastructure Funding initiative. This didn't mature, but I would not be at all surprised if this building were to become involved in studies of how a community's interactions in a complex physical space can be supported more and more deeply by computation. As a very simple example, suppose that you are starting a lecture and the projector is missing; then instead of phoning the caretaker to find it, why not send a message to the projector itself saying "Where are you? or better still "Please arrange that you appear in Lecture Room 1 as soon as possible." (At least the first part of this was essentially done by Professor Andy Hopper's so-called Active Badge project, for locating people, when we lived on the New Museums Site.) The more adventurous possibilities are endless; not the easiest part is working out what the community actually wants.

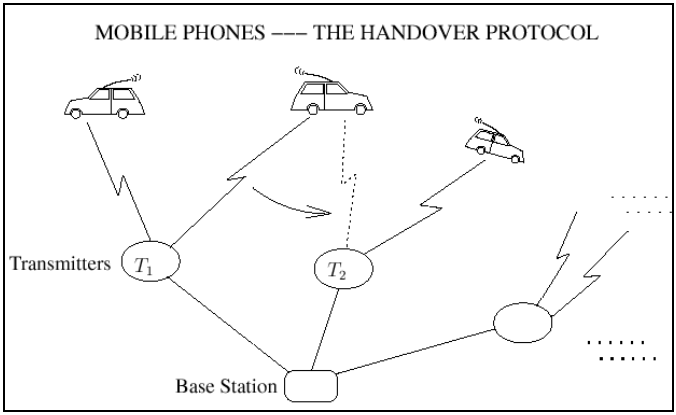
Handover Protocol

Ten years ago, my colleagues and I worked out a so-called Calculus of Mobile Processes, which we called the Pi Calculus, to explore the complex interplay between computation and communication. Later in the lecture I'd like to say more about this kind of work. For now, I'd like just to illustrate what it was

aiming at, by a simple experiment we did with Swedish Telecomm to test out the calculus.

Imagine a mobile phone system (see the picture below); there is a single base station, a number of fixed transmitters around the country, and as many mobile phones as you like. Each phone is connected - using some wavelength - to the transmitter, say T_1, closest to it. When the phone moves the signals get faint, and T_1 alerts the base station of this. After deciding to have the phone connected to another transmitter T_2 , the base station initiates what is called the Handover Protocol. It works like this: the base station identifies a new channel on which the phone will interact with T_2 ; it tells this to T_1, which tells the phone and severs connection with it; the base station also tells the channel to T_2 , which then enters dialogue with the phone.

What the Pi Calculus does is to represent the *potential* interactive behaviour of each agent in the system by a little set of equations; then one can derive equations describing the *actual* interactive behaviour of the whole. Using the mathematics of the calculus, one can immediately exhibit properties of the Handover Protocol; a simple example is an *invarian*t, for example that no phone is ever connected to more than one transmitter. Proving these things amounts to a verification of the Protocol. Of course, this is a simple task. All the same, to verify the same properties for a protocol written in C, say, is rather like using a nut to crack a sledge-hammer …



Nanocomputers

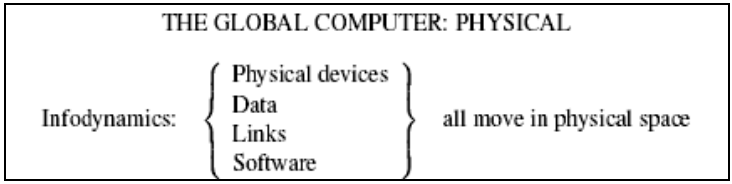
Professor Ehud Shapiro, of the Weizmann Institute in Israel, gave a talk a few weeks ago at the Royal Institution in London about recent work published in the Journal Nature, in 2001. He and his colleagues have built molecular computers out of DNA and enzymes. So far these automata do only simple things, and

slowly; but 10^{12} (a million million) of them can live in a tenth of a litre of water, at room temperature, consuming virtually no energy. His vision is to build a general purpose computer which can be installed - in the plural - within the human body, identifying malfunctions and correcting them. Of course there is a long way to go; in the body's space there would have to be a whole organisation of these machines, coordinating with the body's own mechanism, both for the purpose of diagnosis and to put a remedy into effect.

Shapiro has a second project, related but so far independent, to model biological processes and information pathways. Traditionally, differential equations or Monte Carlo methods have been used; but these do not capture the spatial structure of the processes and pathways. Shapiro is therefore using recent computer science models of concurrent mobile processes, including the Pi Calculus which I have just mentioned, and also the Ambient Calculus of Cardelli and Gordon at the Cambridge Microsoft Laboratory. By adding stochastics he is able to get informative models of such things as signal transduction in cells.

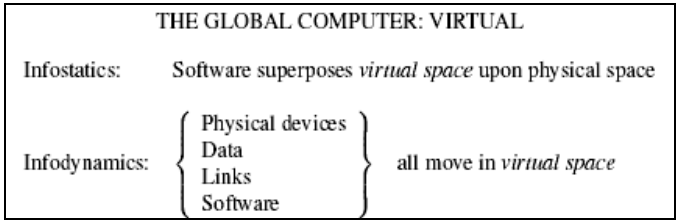
Space and Movement

These three examples - the building support, the Handover Protocol and the biocomputers - show us that communication across space has become inextricably bound up with computing. In a case like the Handover Protocol we can even say that the process, which involves many communications, *actually is* a computation; for an interaction between neighbouring agents is as much a computational primitive as, say, writing a symbol onto the tape of a Turing machine, or fetching an operand from a register in an original stored program computer such as the EDSAC of Sir Maurice Wilkes. With the advent of the Internet, and the worldwide web on top of it, this interactivity has been multiplied a million fold. Quite simply, a *global computer* has come into being. What are the events in this computer? It's all to do with movement, and I shall call it *infodynamic*s.



We have illustrated most of these movements; for example, in the Handover Protocol a link was moved from one transmitter to another. As for software moving, think of viruses, and also the applets that we download, enlivening a website. But as you look deeper, things get more difficult and *much* more interesting. For by means of software - the critical component in all of this – a

*notiona*l, or *virtua*l, space is laid on top of the physical space; that's what the worldwide web does for us! The physical separation of things - whether physical devices, data or software itself – is hidden from us in virtual space, where they may appear to be contiguous. Conversely neighbours in physical space may be totally unconnected in virtual space, like two adjacent mobile phone users on a train. And there is movement in the virtual space too. To go a step further, one can see *all* computation in the global computer as movement, or unending reconfiguration, in this totally new hybrid between physical and virtual space, which I shall call *infospac*e.



This is a lot to swallow! It does somehow reflect our experience on the Internet, even though we don't have to think deeply about infospace when we are exploring it, any more than we think deeply about physical space when exploring *tha*t. But note one detail: this structure is in a sense *universa*l, for every physical entity can be modelled by a virtual one; so parts of the global computer – even the whole - can be modelled, virtually, within it.

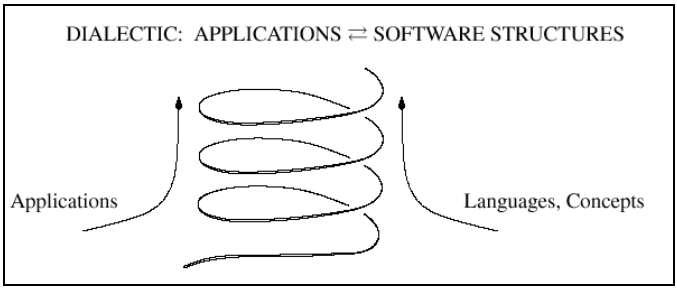
Is it really true that physical devices can move in virtual space? Yes; a hand-held device, on receipt of some password, can enter a privileged virtual *domain* where new interactions become admissible. Can software really move in the virtual space created by software itself? Well, yes; an applet, say, may arrive through physical space and appear in your own virtual workspace. It may then be quarantined within a virtual *firewall* while it is checked out for reliability in some way, and only then released from the firewall; or, for security, it may be kept within a *wrapper* which mediates and monitors all interaction with it. These semi-technical terms - domain, firewall and wrapper - connote some sort of control; this can be interpreted as controlling a spatial region.

Progeny of infospace

We have to realise that the father of this hybrid infospace is network technology; networking is what has brought it to life. Pursuing the analogy, infospace has a mother too - the *software space* which has grown up with the stored-program computer over the past fifty years. This is the space explored by algorithms, data structures and programming languages, which

were in turn brought to life by stored-program computers such as the EDSAC fifty years ago.

It's worth tracing one thread in this growth of software space. This will show that its familiar structures - the data structures and procedure structures of conventional programming – were not delivered along with the idea of a stored program, but actually developed through a cycle of applications, formulation, wider application and so on. Of course, the cycle is really a spiral:

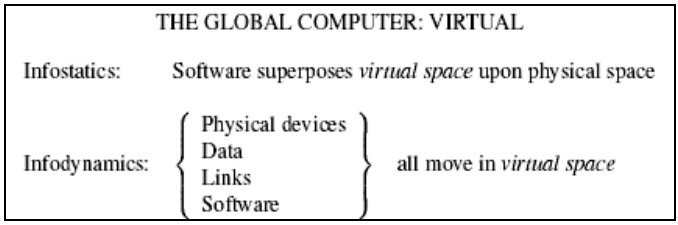


This spiral is well illustrated by so-called *object-oriented programmin*g. There were faint beginnings of this idea even in the language called ALGOL 60, forty years ago. ALGOL 60 had its notion of procedural (or subroutine) structure; for example, a procedure for finding the shortest route between any two given cities, say Cambridge and Birmingham, could be used on successive occasions for many pairs of cities. The feature which ALGOL 60 pioneered was the idea that any procedure could usefully exploit its previous experience. The language allowed a procedure to have what was called its *own variable*s; these gave it a memory from one invocation to another, so that it could act more efficiently on successive occasions. In the case of finding shortest routes, its memory today could recall certain shortest sub-routes which it worked out yesterday for another purpose.

Thus computational procedures achieved real existence as *computing agent*s, by being able to record their past. This notion struggled into the culture incrementally over the next two decades. One challenging application was in the area of computer simulation of real-world processes, like production lines. These simulating agents developed, via the language Simula, into the so-called *objects* of object-oriented programming. They became a new model of software behaviour which had a dramatic influence, both on software engineering and on the further design of programming languages. Note the obvious spiral in mutual development of the model and the real world of applications. There are many other examples of software concepts evolving in this way.

Software Engineering

Before getting up to the present date, and asking what happens next, I want to digress a little, and look at the effect this spiral has had upon the software industry. The rise of the spiral - i.e. the change in how we think about applications and program them - has been dramatic, but the growth of demand for software products, enabled by the advance of technology, has been unprecedented and almost overwhelming. Software houses have been forced to rush products out, in order not to lose the market. Even if they could pause and contemplate the programming models, looking for theories on which to base rigorous methodologies, they would not have found those theories complete.

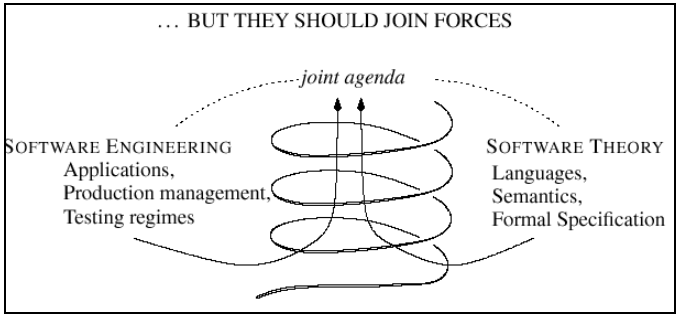


So on the one hand, computational theories have been linked to the spiral via languages and models. In exceptional cases they have influenced applications directly, for example through specification methodologies and analytical tools. On the other hand. software engineering has placed itself firmly on the other side of the spiral. It has concentrated on disciplines for software *testin*g, and upon *management* of the software production process, rather than upon the nature of its raw material, the software. It's a bit like carefully designing the production and testing regime in a car factory, without much attention to the properties of the steel that cars are made of, and the fuel they use.

Furthermore, there has rarely been time to *document* software products in any accurate way. This has led to some amazing statistics. About _five years ago it was estimated that between 80% and 90% of software engineers' time was spent re-engineering old software, now called *legacy* software, to adapt it to new requirements. This re-engineering often consists in poring over millions of lines of old code, trying to find the right way to change it, because the documentation does not help, and - even after the changes - not daring to jettison any part of the 'old' code because it may still get used in a way that isn't evident. Thus the pile of legacy software grows, and presents a bigger challenge for the next time it has to be changed.

Who's at fault, and what to do?
It is tempting, but wrong, to blame companies or academics for this. The fact is, the pressures of opportunity and the market have driven things out of control. Moreover, informed lay opinion seems not to expect that the solution will lie in finding a rigorous science of software. For example, the Economist in its recent Technology Quarterly bemoans the current state of software reliability, but the furthest it goes in discussing remedies is to point to the efficacy of testing-cum-management techniques such as extreme programming or the five-step capability maturity model.

It would be downright foolish to suggest, or hope, that a rigorous scientific model can *replace* a management-cum-testing regime for software. But it is a Grand Challenge for the academic-industrial partnership, over the next two decades, to join a scientific model with software methodology, each informing the other in the way that *applications* and *languages* inform each other - in other words, taking up their position at the centre of the spiral, where they should be:



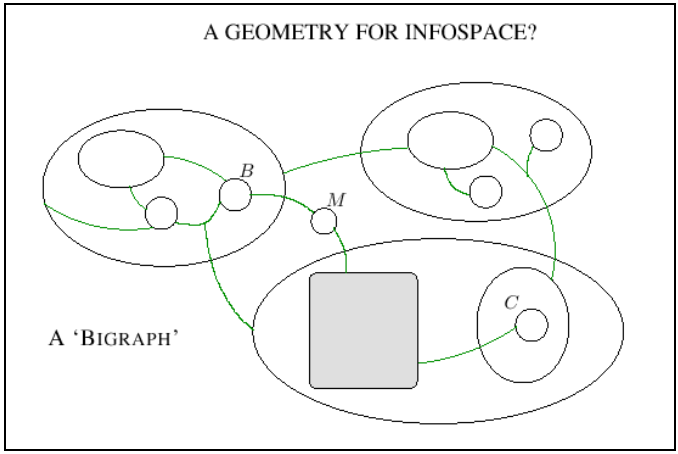
There are two things in our favour here:
- First, the world of web applications, e-business etc is so *different* that old software just can't be used; things are being done in new ways. So if we work quickly enough we may be able to build scientific models before piles of inscrutable new software build up;
- Second, distributed and interactive computing is so *difficult* that we may just be shocked, or forced, into a more scientific habit.

Modelling the hybrid infospace
On that note let us return to infospace or the global computer, and let's ask what an underlying model of such a huge, interactive and spatially distributed system might look like.

The concepts which evolved in software before the worldwide web, especially objects and concurrent processes, go some way

towards a model of the global computer. We have become somewhat familiar, both in our languages and in our models, with handling interactions between processes - via such notions as concurrent threads (say in Java). So there is a temptation to keep these concepts unchanged, but just to add a notion of *place*, or *locality*, which will do for both the physical and the virtual space, together with some way of representing movement among these localities. But this is wrong! Why? Because it violates the principle of Occam's razor, that you should create no more entities than you need. For there are prototypical localities lurking around every corner in traditional software; many of them are reflected in the levels of syntactic structure of our programming languages.

So we would like to weave space into the very basis of our new model. Instead of merely using spatial metaphor to explain or teach a our ideas and theories, let us make the model itself unashamedly geometric. The time is ripe for the closet geometry of computing to, as it were, *come out*. Let me show you what I mean in a diagram or two, representing one approach among several that computer scientists are now proposing.



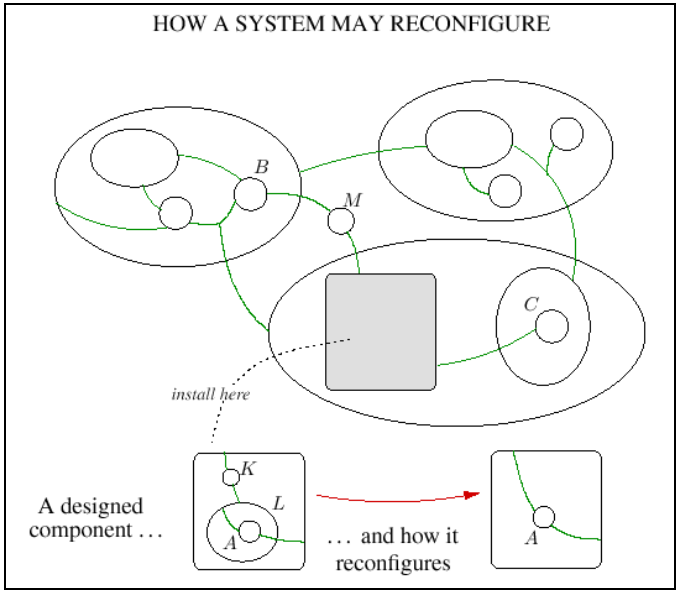
This diagram shows what I call a *bigraph*. A single bigraph, billions of times larger, would represent the state of the global computer at one instant, so the bigraph at the next instant would be different. The bigraph, and the way it changes from instant to instant, is the whole story of infospace. Primitive software actions are represented in just the same way as movements of devices in physical space are represented. These, and only these, are the computational events.

Details of bigraphs

The stuff of a bigraph is its *nodes*; the ovals and circles. A node may be a city, a building, a computer, a hand-held device, a website, a message, a piece of software, a piece of data - in short, anything that may have physical or virtual location. It is called a bigraph because there are *two* structures on these nodes; the *topographical* or *spatial* structure, represented by the way the nodes are nested, and the *connectivity* represented by the thin lines. These are totally independent: `Where you are does not effect who you can talk to'.

This isn't just the space *within which* computing occurs; computing *consists in* reconfiguration of this space. In this picture, a node could represent a *physical device* entering a domain of control, a *virus* doing the same thing, or a *number* about to be operated on. What happens in the machine is dictated by rules, which may vary from place to place, saying that a certain configuration can change into another one.

As a simple example, suppose you are going to design a piece of the system to fit in the blank grey box. It may be a simple piece of message routing software; think of *M* being a message from *B* in MIT to *C* in the Cambridge Computer Lab. Here is one possible thing your little subsystem might do:



Your subsystem, when installed, will allow *M* to turn the key *K* ; this will unlock the lock *L*, giving access to an administrative agent *A* that will in turn purvey *M* to *C* . (Of course, when you build your system, you also define the reconfiguring powers of keys and locks.)

This is a trivial example. The essential point is that substantial reconfigurations will be modelled in the same way (just as a tiny program is made of the same stuff as one with millions of lines). In this example the reconfiguration in a *local* one; it involves only entities that are neighbours in the topography. But a reconfiguration need not be local; it might consist of synchronized change involving widely separated parts, connected by the thin lines. This is the fiction of *action at a distance* which the web and the Internet allow us to entertain.

This points to a crucial feature of the models we are looking for. The stuff of the model must be able to represent both the fiction or *abstraction* which is entertained by a participant (say a human participant) as well as the *reality* by which this fiction is engineered. My colleague Peter Sewell points out that it is only through representing both the abstract and the concrete with the same modelling stuff that one can hope to verify or refute the claim that the concrete does indeed *realise* or *implement* the abstract. In fact, with a model of this kind, his group have recently verified some protocols for communication between mobile entities that come close to those actually used in the Internet.

One further point before we leave this model. Any configuration of the Internet and software running on it will be vast; coping with an abstract as well as a concrete representation is a necessary device to cope with this scale of things, but it's not the only one. We need to analyse the behaviour of *fragments* of a bigraph. For example, we need to be able to say that there will be, or won't be, any noticeable difference if one fragment is replaced by another with the same connectivity interface. (For example, the fragment in the top right corner of the picture might represent the hardware and software of an Internet banking service; the bank might want to replace it by a system using not a single computer, but six of them linked by local area network.)

Ideally, we want to know if the environment can observe any difference in behaviour after such a replacement. To formulate this question accurately, and then to answer it, we have no choice but to refer to a *mathematical* notion of the *behaviour* of a fragment, defined by how it interacts at its interface - *not* by how it looks if you take it apart. Suitable behavioural theories have been developed by computer scientists over the last twenty years, often needing new mathematics and logics. There are signs that they can be extended to a geometric model such as I have

shown you, and I regard it as a big part of the Grand Challenge to push this work forward.

Challenge for the Computer Lab

To summarise: We are faced with the extraordinary task of understanding what is probably the most complex artifact in human history, the global computer. Only by understanding it well can we expect it to serve us well. But at present it is developed by software methods which - however successful - rely upon *no previously developed scienc*e. Compare this with any other standard engineering discipline; structural engineers, for example, can rely on material sciences and ultimately upon physics.

A CHALLENGE FOR TWO DECADES

Build a scientific model of the GLOBAL COMPUTER, in collaboration between engineers and theorists, providing one conceptual frame for

- *describing* the whole,
- *prescribing* the parts we build, both hardware and software,
- *designing* those parts in a way that is seen to match the prescription, and
- *modifying* parts whose prescription is modified.

The global computer gives us both opportunity and tremendous incentive to develop a theory and a new style of software construction and adaptation which are aspects of one and the same scientific model. The theory will be *descriptive* of all that happens in the global computer, while the software will be *prescriptiv*e, determining the parts that we build; they will be expressed in the same terms. To get this to happen certainly involves half-baked prototypical models of the kind I showed you. But to succeed, it must involve experts in communications networks, programming languages, security, interface specifications, special hardware, machine assisted reasoning, and more. It will only come about through the normal process of scientific advance, and will probably take two decades. It's a much bigger aspiration than, for example, a five-year programme managed by a research funding council.

Who will do it? I fervently hope that the Computer Lab can play a large part. We have many of the kinds of people it needs. Also, thanks to the Gates Foundation, we have the opportunity to do our science and engineering under one splendid roof. Far from splitting our theories and practices apart, we can use this challenge to bring them closer together than they ever have been.

Iris Recognition Seeing International Deployments

By

John Daugman

Automatic visual identification of persons has been a long-standing goal of artificial intelligence. Algorithms developed at the Computer Lab for this purpose are now being deployed internationally at airports and other facilities where reliable and rapid identification of persons is required. The algorithms are based on a combination of diverse mathematical ideas that allow the encoding and recognition of the complex random patterns that are visible in the iris of a person's eye from some distance. The UK Home Office now has authorized the use of these algorithms to replace passports, for automatic entry of trans-Atlantic travellers into the UK at Heathrow, as have the Dutch border police at Schiphol; and the Canadian Government has mandated the same technology for all 11 of Canada's international airports. A number of governments, including the US and UK, have floated consultation papers proposing possible use of these algorithms as the basis for a national ID system, and for the enablement of foreign visas and travel documents.

There is a long tradition of regarding eyes as "windows to the soul." For example in Shakespeare's *The Merchant of Venice*, Portia's suitor Bassanio recalls how "Sometimes from her eyes I did receive fair speechless messages." Eye contact is a crucial aspect of social interaction which we scrupulously (if unconsciously) monitor and calibrate, especially between the genders; giving either too much or too little of it is loaded with significance. (Perhaps less so among computer scientists.) In addition to gaze monitoring, since we seem to realise instinctively that even the diameter of the pupil is related to emotional state and arousal, Italian women of the Renaissance cosmetically applied atropine to their eyes to enlarge their pupils. For this reason, charmingly, the natural herbal source of atropine is a plant named *Bella Donna*. But beauty aside, the eyes can serve as reliable windows to a person's identity because of the great complexity and randomness in the visible patterns of the iris. All published reports from independent testing labs, such as the NPL, that have been running trials of my algorithms for iris recognition have reported getting zero false matches even after several millions of test iris comparisons.

The key is randomness and complexity. Randomness plays a crucial role in many sciences, and increasingly also for technologies. In biology, random variation by mutation is the engine of evolution; in physics, random state variables are key to quantum mechanics and to thermodynamics; and in the information sciences, random sequences are central to theories of

cryptography, data compressibility, and algorithmic complexity. Many methods exist for measuring and describing the randomness of variables or patterns. Perhaps the most interesting of these are the proposals by the Russian mathematician Kolmogorov, that the complexity of a random sequence is equal to the length of the shortest program that can generate it, and that a pattern is defined as *algorithmically random* if it is its own shortest possible description.

Biometric identification systems all rely on forms of random variation among persons. Examples of deployed biometric identifiers include fingerprints, hand shape, facial appearance, retinal vein patterns, and DNA. Many people assume that DNA is the "ultimate biometric;" but of course DNA does not distinguish identical twins (so its error rate is at least 1% across a population at large), nor do DNA tests reliably distinguish persons who are just closely related. And by requiring a physical sample, it is neither real-time nor unintrusive. (In its defense, however, it must be admitted that approximately one-half of the human population is constantly trying to provide a sample of its DNA to the other half.)

Almost the entire science of pattern recognition can be reduced to a single issue: the relation of within-class variability to between-class variability. Patterns can be reliably recognized only if the variability among different instances of a given class is smaller than the variability between classes. Another way of saying this is that the clusters in some multi-dimensional space should not overlap: their spacings should exceed their diameters. In our context the classes are individual persons, seen at different times and in varying circumstances. We wish to represent them in such a way that they are more different from each other than the degree to which each can differ from itself. The key to the variability between classes is the amount of randomness they incorporate. Having more dimensions of independent variation creates signatures with more uniqueness. But while seeking to maximise between-person variability, biometric templates must also achieve minimal within-person variability across time and changing conditions of capture. In the case of face recognition for example, difficulties arise from the fact that faces are changeable social organs displaying a variety of expressions, as well as being active 3D objects whose projected 2D images vary with pose and viewing angle, illumination, accoutrements, and age. Against this within-person (same face) variability, between-person variability is limited because different faces possess the same canonical set of features, always in basically the same canonical geometry. As is easily proven, among frontal images of any given face, the variability even just from illumination angle alone can be much larger than the variability among images of different faces captured with fixed expression; and it has been

shown that for images taken at least one year apart, even the best face recognition algorithms have error rates of about 50%.

For all of these reasons, iris patterns offer a much more powerful means for reliable visual identification of persons when imaging can be done at distances of about a meter or less, and especially when there is a need to search very large databases. (For any recognition system, the error probability goes up almost in proportion to the number of alternative hypotheses that must be entertained.) Although small and sometimes problematic to image, the iris has the great mathematical advantage that its pattern variability among different persons is enormous. In addition, as an internal (yet externally visible) organ of the eye -- indeed the only internal organ of the body that is externally visible -- the iris is well protected from the environment, and stable over time. Of course, there is a popular occult belief in systematic changes in iris patterns reflecting the state of health of each of the organs in the body, one's mood or personality, and revealing one's future. Practitioners skilled in the art of interpreting these aspects of iris patterns for diagnosing clients' health, personality, future, and mutual compatibilities, are called iridologists. Belief in iridology is popular in Roumania and around the Bay Area in California.

The iris begins to form in the third month of gestation and the structures creating its pattern are largely complete by the eighth month, although pigment accretion can continue into the first postnatal years. Its complex pattern can contain many distinctive features such as: arching ligaments, furrows, ridges, crypts, rings, a corona, freckles, and a zigzag collarette, some of which may be seen in the iris image in Figure 1.

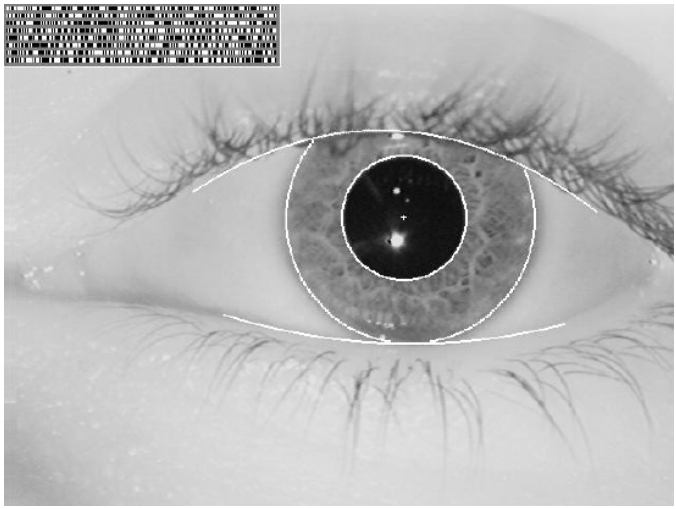


Figure 1: An iris with its phase-sequence "IrisCode," for automatic identification of persons.

The morphogenesis of the iris pattern is chaotic, determined by the random adhesion of pectinate ligaments (connective tissue) during development. The pattern is epigenetic: its details are not genetically determined, although of course eye colour is genetically determined. (Every biometric lies somewhere along a continuum between genotypic, such as DNA sequence, and phenotypic, such as iris patterns or fingerprints. Facial appearance is midway in between, having a large genetic component as revealed by identical twins, but also a large developmental dependence as revealed by changes over time.) The absence of genetic penetrance into the detailed patterns of the iris is profound: there is no correlation whatever even between the left and right eyes of each person. The distribution of Hamming Distances between genetically identical irises (see my website) is statistically indistinguishable from that between genetically unrelated eyes. One implication of this observation is that human clones, in the inevitable future, will remain fully distinguishable from their genetic sources and from their brethren by their iris patterns.

I encode the detailed pattern of an iris using complex-valued 2D wavelets. These are relatively new mathematical beasts which allow one to speak of the phase of a signal or pattern at any point, a notion that previously would have made sense only for simple harmonic functions such as sinusoids. Wavelet analysis resembles Fourier analysis but with the restoration of locality, which is completely lost in a Fourier representation. A wavelet

encoding of an iris pattern extracts its phase sequence. This idea is summarised in Figure 2.

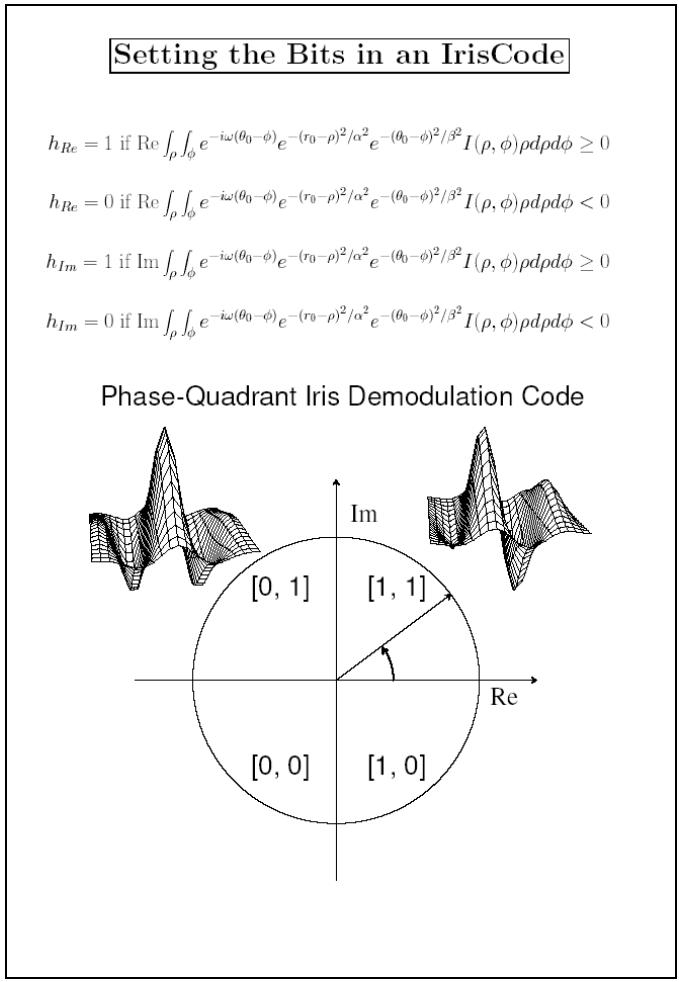


Figure 2: An IrisCode is computed using 2D wavelets that demodulate the iris pattern into a unique phase sequence.

A quadrature-phase pair of 2D wavelets is plotted here, and the equations give their mathematical form. The classical operation of projection between two functions is performed, to project each region of the iris pattern onto a complex-valued, quadrature phase pair of wavelets. This generates a complex number, whose real and imaginary parts allow one to speak of the phase of the iris pattern at each point, and thereby to encode the entire iris pattern as a sequence of phases. I quantise each computed phase value to the nearest quadrant of the complex plane, as illustrated at the bottom of Figure 2, thus specifying two bits of phase information for each number. In this respect an "IrisCode" (shown as a bar code in Figure 1) resembles a DNA sequence, if one regards the four quadrants in Figure 2 as analogues of the four nucleic acids. Altogether an IrisCode comprises 2,048 bits, or 256 bytes, of such phase information, whose sequence becomes a code for an individual's identity.

When searching for a person's identity in an enrolled database using their iris pattern, comparisons against all stored IrisCodes are performed to see if any of them match. This is an extremely rapid process: on standard low-cost hardware the search speed exceeds 100,000 persons per second, and of course it can be parallelised for larger databases. IrisCode comparisons simply require taking the bitwise Exclusive-OR between the two complete bit vectors and measuring the resulting norm, which is their Hamming Distance. This distance between different irises is distributed exactly as a binomial, because each bit comparison is a Bernoulli trial ("coin toss"), and sequences of Bernoulli trials even if correlated generate binomial distributions. A histogram of 9 million iris comparisons is shown in Figure 3, together with the theoretical binomial (solid curve) which fits the data perfectly. The smallest Hamming Distance observed between two irises among these 9 million comparisons was 0.334, which means that one can tolerate up to about a third of the bits disagreeing between two IrisCodes and still declare them to be a match, with negligible probability of error. This is a remarkable degree of tolerance for data corruption. The key to being recognised by your iris pattern is simply that you fail a test of statistical independence against your own (previously enrolled) IrisCode; but you are statistically guaranteed to pass that test against all other IrisCodes. All of those alternative hypotheses about your identity are rejected as decisively as concluding that a sample value below 0.33 was not drawn from the distribution seen in Figure 3.

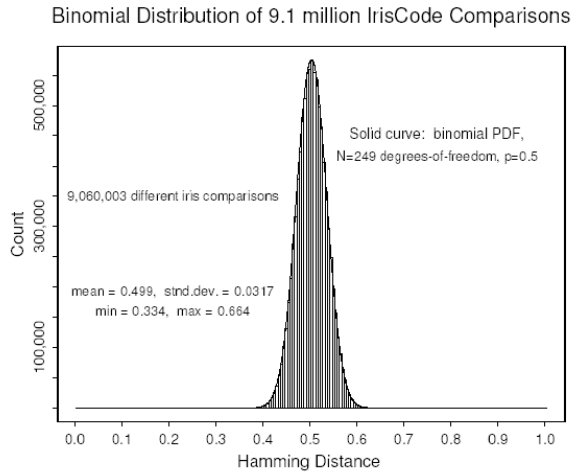


Figure 3: Binomial distribution of 9.1 million IrisCode comparisons. Tests by independent laboratories have so far produced no False Matches. The likelihood of two different IrisCodes being deemed a match by chance is the same as tossing a fair coin 250 times and getting less than one-third "heads."

The algorithms that I've described here are running in all currently deployed iris recognition systems worldwide, as licensed executables. Special purpose cameras embedding this process are sold by companies such as Panasonic, Oki, and LG. My focus has shifted now to algorithms for identifying the scribes of medieval manuscripts by their handwriting (a kind of behavioural biometric!). But an unexpected application of the iris algorithms recently arose when National Geographic magazine asked me to try to identify an unknown Afghan refugee girl whom they had photographed in Pakistan in 1984, who left her home after the Soviet occupation of Afghanistan, during which both of her parents had been killed by Soviet bombing. Her photograph on the cover of the magazine in 1985 became the most famous and most widely reproduced photo in the magazine's 100 year history. In 2002 National Geographic came to the Computer Lab with a photo taken recently of a 30 year-old Afghan woman named Sharbat Gula, and asked me if my iris recognition algorithms could tell whether she was the same person as the nameless illiterate 12 year-old refugee girl they had photographed in 1984. Despite the passage of 18 years, the algorithms returned a decisive match for both of her eyes. National Geographic published this result, and launched their "Afghan Girl's Fund" to assist the education of Muslim girls in cultures that discourage the education of girls.

Finally, remaining in Afghanistan: In case anyone is interested in having them, I have been able to compute, from the famous close-up facial video footage filmed in a cave in Afghanistan, both of the IrisCodes of Osama Bin Laden. Readers may be interested to learn that he wears soft contact lenses.

Business Hall of Fame

We are compiling a list of businesses spawned by Lab people and their research. We hope that these may act as an encouragement and inspiration to students and graduates. To add a company to the list please email Jan.Samols@cl.cam.ac.uk.

Ray Anderson
Bango.net (f.1999)
Provides technology that enables paid for content on the Mobile Internet

John Bates
APAMA (f.1999)
Provider of unique software technology – the Apama Engine - which works by continually monitoring streams of data for complex patterns of events, and providing real-time alerts when a match is found.

Richard Boulton
Lemur Consulting (f.2001)
Providers of Information Retrieval technology

John Brimacombe
nGame (f.1997)
Provider of innovative multi-player gaming experiences on a wide variety of new platforms

Jobstream Group plc (f.1993)
Provider of software solutions to meet business requirements of trust and investment management companies, private client departments of banks, accountancy and stockbroking houses as well as a wide range of other client services organizations

Juanito Camilleri
MobIsle Communications Ltd (f.1999)
Malta's second mobile communications company and a wholly owned subsidiary of Maltacom plc.

David Colver
Operis Group plc (f.1997)
Provides specialist financial advisory & capital raising services, specialist financial modelling services and financial model training services.

Ben Coppin
Envisional
Intellectual property protection and monitoring solutions

Peter Cowley
Camdata (f.1984)
Designs and manufactures computer equipment for difficult environments e.g. oil-rigs, power stations etc. Also designs and manufactures networking electronics for security uses.

Paul Cunningham
Steev Wilcox
Paradigm Design Systems Limited (f.2001)
Develops software tools for low power chip design

Robert Darwin
Digital Mail (f.1991)
Provider of communications services

John Daugman OBE
Iridian Technologies (f.1990)
The world leader in research, development and marketing of authentication technologies based on iris recognition

Nicky Dibben (nee Sutton)
Invention Marketing (f.1999)
Provider of strategic and tactical marketing services

Matthew Faupel
Micropraxis Ltd (f.2001)
Computer consultancy

John Fawcett
Matthew Parkinson
Andrew Rice
Invest Solutions Limited (f.2002)
Provides flexible solutions to website needs

Ben Finn
The Sibelius Group(f.1993)
World's leading music notation technology provider

David Greaves
Tenison Technology EDA Ltd (f.2000)
Codesign software specialising in C and Verilog interworking

David Greaves and Andy Hopper
Virata (f.1993)
Provider of communications processors combined with integrated software modules to manufacturers of equipment utilizing DSL technologies. Now called globespanvirata.

Daniel Hall
Adrian Wrigley
ART (f.1995)
Leading providers of
Photorealistic rendering solutions to the 3D industry

Demis Hassabis
Elixir Studios
(f.1998)
Leading developer of interactive entertainment software

Andrew Herbert
APM Ltd (f.1985)
An independent company specialising in the technology and application of networked and distributed IT systems

Digitivity (f.1996)
(APM spin-out)
Applet management system which provides automated security by running all applets outside the firewall to ensure they never compromise network security.
Acquired by Citrix Systems Inc 1998.

Andy Hopper
ARM (f.1990)
Industry's leading provider of semiconductor intellectual property.

Telemedia Systems Ltd – now Internet Pro Video (f.1996)
Leading supplier of browsing technology to the professional video market

Mike Kemp
Sintefex Audio
(f.1997)
Dedicated to innovative research, development and consultancy in relation to digital audio technology and manufacturing of digital audio products for the professional user.

Tim King
UK Online (f.1994)
The UK's first full ISP
Sold to EasyNet 1996

Jack Lang
Resigned as Chief Technologist of ntl to found
Interactive Digital Television Ltd (f.2001)

NetChannel Ltd (f.1995)
Interactive TV business
Acquired by ntl

Electronic Share Information Ltd (f.1993)
Online provider of financial data
*Acquired by E*Trade Ltd*

Midsummer House Ltd (f.1998)
The first restaurant in Cambridge to receive a much-coveted Michelin Star

Jack is also a Trustee of The Design Trust, a registered charity which promotes the excellence of British design and helps designers with business training after they leave college.

Ulrich Lang
ObjectSecurity Ltd (f.1999)
Provides specialist consulting in IT security

Ian Leslie
Nemesys (f.1994)
Producer of hardware and software to allow cost effect video connection to ATM networks.
Acquired by FORE systems 1996

Ian Leslie
Simon Crosby
CPLANE (f.1999)
Network control software products

Stephen Love
OptionExist (f.1993)
Provides innovative product development and technical consultancy services

Chris Oswald
Gareth Williams
Equisys (f.1987)
A leading provider of business communications solutions
Awarded Queen's Award for Export 1999

Martin Porter
Muscat Ltd (f.1995)
Information retrieval systems
Sold to The Dialog Corporation

Damian Reeves
Adam Twiss
Zeus (f.1995)
Leading web server technology developer

Robert Sansom
FORE (f 1990)
A leading designer and producer of high performance networking products based on ATM and IP technologies.
Sold to Marconi (formally GEC plc) April 1999 for $4.5bio

Quentin Stafford-Fraser
Stewart Lang
Ellipsian Ltd (f.2002)
Exploring ways to bring technology-based solutions to bear on real-world problems and to make interesting connections between business needs and University expertise

William Tunstall-Pedoe
Genius 2000 Ltd (f.1998)

Adam Twiss
Saviso Group (f.2002)
Provides strategic and operational assistance to early stage technology businesses

Nicko van Someren
Ncipher (f.1996)
Leading developer of hardware and software internet security products

Tim Ward
Brett Ward Limited (f.1996)
Software and systems engineering

Computer Laboratory News

New Staff

Tim Harris has joined the permanent staff of the Computer Laboratory as a new Lecturer in the Systems Research Group, teaching the Part 1B course on Concurrent Systems and Applications which covers concurrent programming in Java and transactions. Last year he finished his PhD on Extensible Virtual Machines which allow programmers of a JVM-like environment to exercise similar low-level control over their code's execution to that available to C programmers. His current research interests cover lock-free data structures (intricate designs of lists, heaps, and the like that work correctly on multi-processor machines without ever needing to use locks), xenoservers (a public infrastructure for distributed computing) and debugger support for multi-threaded and distributed systems.

Computer Lab honoured

The William Gates building is one of 58 new buildings across the UK and EU to have been honoured by the Royal Institute of British Architects (RIBA) for high architectural standards and for making a contribution to the local environment. The laboratory, which was designed by London architects RMJM, will now be on the longlist for the RIBA Stirling Prize, the shortlist for which will be announced on 12 September.

The University Alumni Weekend

This year's events will take place over the weekend of September 27. The Computer Laboratory will be taking part on Sunday 29 September when Stephen Allott, Director of Development, will be giving a tour of the impressive award-winning William Gates Building. If you would like to sign up for the tour please contact the University Alumni Office at 10 Trumpington Street, Cambridge, CB2 1AQ tel 01223 332 288.

Graduates in the News

We would welcome news of any appointments, distinctions gained or honours and awards made to graduates of the Laboratory. Please contact the Cambridge Computer Lab Ring office

Starting a new business and need help? Contact the University of Cambridge Entrepreneurship Centre

The University of Cambridge Entrepreneurship Centre (CEC) has been set up to help University students, researchers and faculty develop their business ideas. The Centre's activities are divided into three areas:

1. Teaching and Training: educational activities to inspire, build skills and embed through training

2. Business Creation: advice and mentoring of new ventures and entrepreneurs at early stages of development

3. Research: Best research on all the major areas of learning relevant to the support of successful knowledge-based entrepreneurs

CEC has offices in the centre of Cambridge (4a Trumpington Street) and at West Cambridge (within the William Gates Building).

CEC also works closely with the other parts of the University to provide specific expertise to help the development of new business ventures. These include the University's Technology Transfer Office (advice in intellectual property), University Challenge Fund (seed investment in new University ventures) and Cambridge University Entrepreneurs (business plan competitions).

For further information on ways in which CEC can help you develop your business ideas, please see www.cec.cam.ac.uk.