# *Preface*

## Background and Motivation

When setting out to write this book the initial target was to produce a relatively straightforward introduction to basic computer graphics algorithms. As work has progressed the goal posts have moved! This is partly because there were already many excellent books covering this aspect of the subject, but partly because there seemed to be other important approaches to the subject that needed to be addressed beyond merely how to automate the production of drawings and pictures.

The evolution of information processing technology and its deployment is still under way, and the role of computer graphics within it is being extended and modified year by year. Though it remains necessary to present the technical side of the subject in order to understand the constraints that limit its use on one hand and establish the potential it has on the other: it now seems important to present graphics as one among a series of modelling techniques that can be integrated together using computers, in a way that makes a far better use of the new technology than merely automating what was previously done.

The subject of computer graphics draws on results from a variety of disciplines: the properties of analogue and digital electronics, and even device physics that allow computing and display devices to work; the computer science topics that cover the computer languages, operating systems, data-structures and algorithms needed to program these systems; the perception studies that range from the experience of the artist to the scientific experiments employed in perception psychology that underpin the way interactive and animated displays can be designed: are all important but they only make up part of the subject. There are also all the application areas that create their own specialist demands and contributions to the subject: cartography, engineering and architectural design, scientific visualisation, medical imaging, robot control, as well as a full range of educational and entertainment applications. This spread of relevant topics by itself poses a presentation problem, let alone providing the background needed to make intelligent or creative use of material they offer.

The contribution made by these disciplines to the subject of computer graphics can be grouped under various headings. A useful classification scheme is that given in Figure 1. This presents three axes, the vertical one being concerned with representing information. This covers modelling types and their structure and properties: mathematics, both computing and natural languages, analogue systems and graphic and scale modelling systems. The applications of these representational schemes to the real world, based on experiments and empirical observation, essentially cover scientific research and its topics can be laid out along the B axis. Finally axis C represents activities that apply the results from A and B to the design and production of new goods and services.

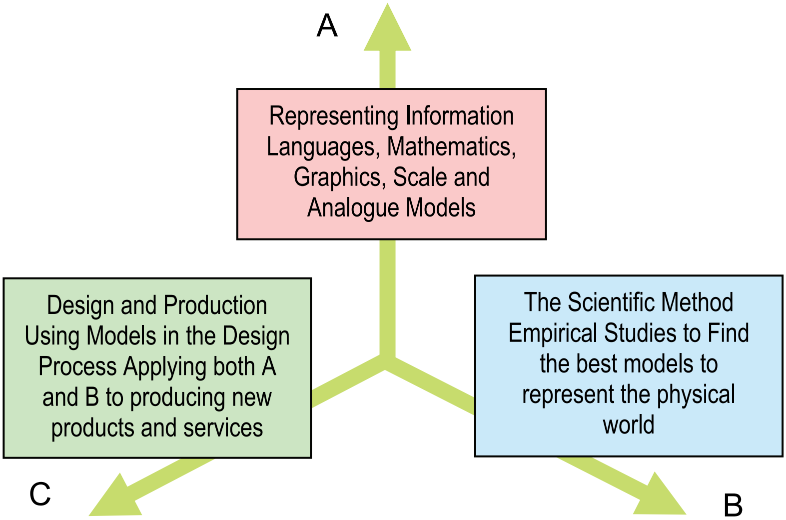


**Figure 1** A modelling classification space

Clearly the various disciplines listed above occupy different places in this classification space. Although the treatment of the subject of this book would seem to lie along the A axis, the decision to present it from a programmer's perspective, moves the approach towards the C axis. Designing and producing programs to support graphic and other modelling systems will need to draw on results from both A and B, but will be a study of how algorithms can be implemented in a working system as much as a study of the mathematical properties that make them possible.

The approach adopted in this book is therefore to treat the subject as a design and modelling task. The design process can be presented in a variety of ways. Computer science is rediscovering and renaming processes well known in other older design disciplines. In particular a graphics-modelling scheme with an associated language called the Universal Modelling Language UML, is being developed to help design large software systems. The problem this addresses is the difficulty seeing structure in thousands of lines of complex programming code: more to the point managing the choices and changes that are integral to their design and development work. A simple schematic for the design process is given in Figure 2. An important new term for an old idea is that of "refactoring", which covers the tasks involved in design integration and optimisation

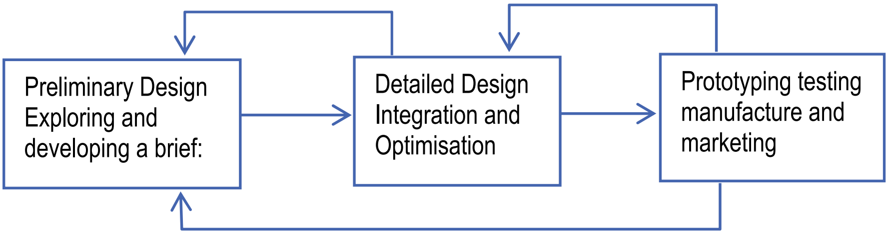


**Figure 2** The design process

The starting idea was to present the design and implementation of a basic two and three-dimensional graphic modelling system. This would allow related topics to be examined only in the depth needed to make design choices, but this still presented difficulties. If existing graphic libraries are examined it is clear that in order to provide a fully working system there is a lot of code which is there to provide error-checking, diagnostic messages, recovery routes for example if *"undo"* commands are provide – all very necessary in interactive environments but obscuring the structure of the key modelling operations. Consequently a compromise approach has evolved. The first two stages, in Figure 2, are explored for important display and modelling tasks but some of the work is only presented as "work-in-progress". This allows key issues that affect the design and programming choices to be highlighted and examined that might otherwise be overlooked, but it means that some of the code and examples are at different levels of development and in some cases all that is done is to show that links between the more detailed studies are possible.

## Setting a Design Brief

The first step in any design study is to explore the context for the new product. This includes the environment that determines the service the new system or product is expected to provide, but also the technical constraints, arising from materials, expected costs, and the speed or size of available components, which may affect the performance that can be provided.

Traditional graphics is divided into many camps, depending on the medium employed. Line drawing, engraving, watercolour painting, oil colours, fresco painting all require considerable skill to master and therefore tend to separate out as independent areas of expertise. There are further subdivisions depending on the preferred subject matter: high art, portraiture, cartooning and many more. Research is being carried out to allow graphic display devices to simulate the colour and texture effects produced by traditional media such as watercolour or oil colour paints. Not only does the use of computer graphics offer a unifying force to this area, but it also links in sculptural art forms and three-dimensional shape modelling. Underpinning computer graphics are various forms of spatial models based on the mathematical representations of geometric-relationships. These modelling schemes also support numerically controlled machine tools; laser lithography and related techniques, and robot controlled manufacturing cells, and potentially provide a unifying link between two dimensional graphic models and fully three dimensional models. In order to cover the full range of possibilities for new systems it seems necessary to place the use of graphics within this larger evolving framework of compatible computer-based facilities.

## The Context: Computer Based Information Processing Systems

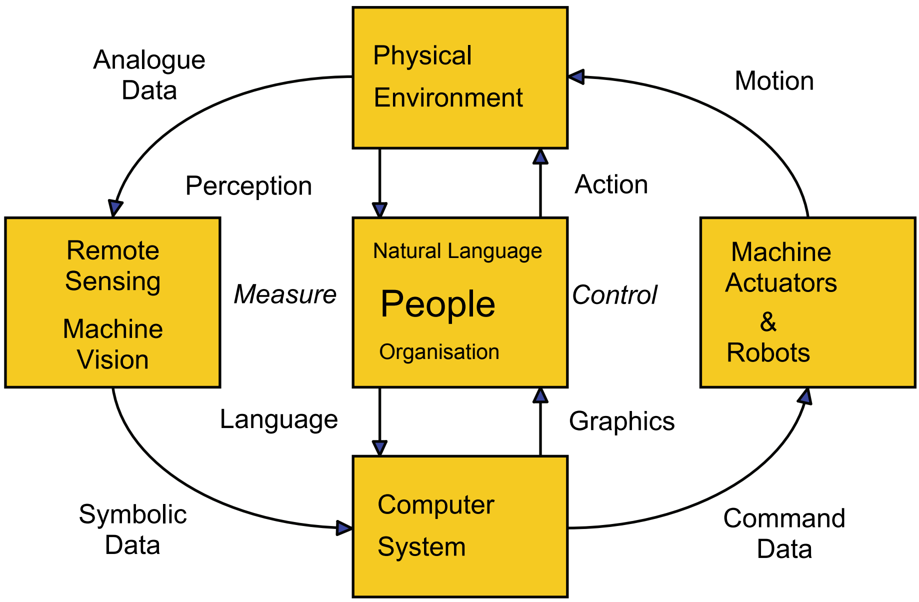


**Figure 3** The place of graphics in future computer based systems

As a change in basic technology, automating information processing systems fundamentally affects the cost, speed and size of many processes and products. This in turn rearranges many existing patterns of use and in turn employment, removing some ways of earning a living but creating many new ways to replace those that are lost. To provide a context in which to understand and predict the consequences of these developments, it seems necessary to study the flows of information that support the essential activities in the system as well as the activities that are being modified by the changes. This involves identifying those tasks traditionally carried out manually that are now more effectively automated, those where human capabilities are still essential, and perhaps most important of all, in a way, the form of the communication interface between the two, needed to maintain an operational and efficient working system. The diagram in Figure 3 gives an abstract framework for this kind of information flow analysis.

Computer systems, which generate more work than they remove, for the same result, clearly are not worth developing. A variation on this observation was a criticism of the San Francisco Bay Area planning simulation model: that it would take more people to collect the data to feed the model than the model in action would service. This can be contrasted with many practical and successful remote-sensing systems where data capture is automated.

One explanation for the desire of governments to issue electronically read identity cards to people is that the automatic logging of transactions such as occurs in supermarkets provides the kind of human activity data that would allow economic planning simulation models to become more practical. The fear of Big Brother is clearly a sensitive political issue, here. A balanced flow of information, or the data that carries it, has to be maintained without bottlenecks if efficient but practical working computer-based systems are to evolve successfully.

Another framework that is necessary to consider when working with information processing applications is the level at which information is represented: the nature of the data. The input of image data in Figure 3 to the computer system is at a completely different level to the language data input by human beings. As they stand they are incompatible without human intervention. The development of machine vision algorithms is necessary if this divide is to be crossed automatically. Figure 4 outlines the layers in information processing systems that have evolved or are evolving as the technology expands and matures and diffuses into everyday usage.

Programmable Hardware
+ Language translators & Operating Systems →
Computer Systems
+ Communication & Parallel Processing →
Networked & Distributed Systems
+ Remote Sensing & Automatic Data Capture & Storage →
Information Systems
+ Feedback & Control →
Management & Control Systems
+ Mobility & Articulated Motion →
Robots & Autonomous Systems
+ Manufacture & Repair →
Self-Replicating Autonomous Systems

**Figure 4** Computer Based Information System Hierarchy

Finally the third contextual framework, which needs to be explored, is the communication between people and machines within computer-based information systems. In early systems the mismatch between the speed of central processing units and data entry led to the creation of multi-tasking and then time-sharing systems. For human beings the eyes provide a very fast input for information. This allows dangerous situations to be recognised and avoided, it allows body language to be interpreted, and signalling using hands and arms to be responded to in real time.

More sophisticated physical forms of communication that require the use of the eyes: such as drawings, diagrams, maps and pictures, all take time to construct and are not generally the basis for real time interactive communication. In contrast hearing has supported the development of language forms of communication. Speech and conversation clearly support real time interaction.

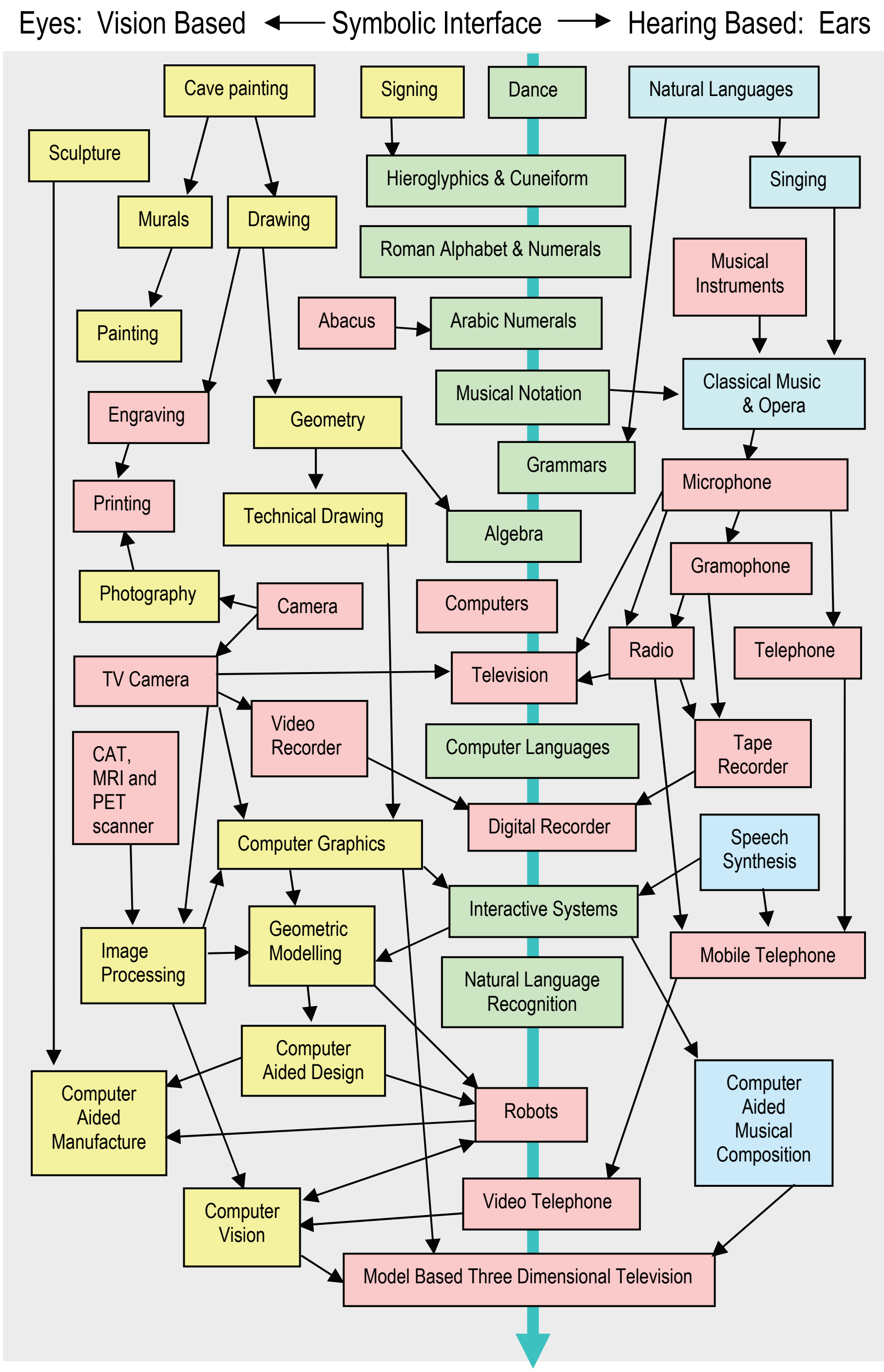


**Figure 5** Evolving communications: Vision v Hearing: Graphics v Speech

Speech is ephemeral. People need to remember what has been said for its content to have any lifetime. In contrast graphics and text are physical artefacts that last for thousands of years, and as long as there are people who know how to interpret them, provide a long term memory or storage for the information they contain, particularly from generation to generation as the basis for education, maintaining cultural systems, and providing a form of memory for the whole community.

The evolution of the relationship between these two modes of communication based on hearing and seeing can be laid out against a rough time line in the way shown in Figure 5.

The ability to create animated images in real time from symbolic language inputs and the automated translation of spoken to text forms of language-input introduces a new "threshold of practical usage" for graphic modes of communication which is only now beginning to be explored. Similarly the development of machine vision systems that can create higher-level models of the environment and bridge to language levels of communication again offer a new "threshold of practical usage" for systems such as TV broadcasting, particularly as they become digitally based. The transfer of storage from graphics and text forms to computer memory systems itself offers new possibilities since this data is physical and can have a long life, but can be turned from passive storage forms into active usable forms without necessarily involving a human agent to interpret it.

These new options cannot be developed by considering computer graphics in isolation. All the related modelling schemes for representing the real world and any virtual worlds under discussion need to be brought together. This is again too large a brief for one book. Consequently the idea is to present graphics algorithms as the primary theme but explore it as a thread through a wider domain.

As an analogy consider prospective developments to be an archaeological site. It is possible to cut "trenches" across the site to get a broad view of what is hidden and it is also possible to dig "pits" more carefully examining material in greater detail where key finds appear. The main story line from chapter to chapter can be considered to be a linking "trench", while the separate studies in each chapter correspond to the more detailed studies in the "pits"! This reflects the hope that an understanding of the overall potential can be provided, but acknowledges it leaves "unexplored" regions that could still contain important material and even contain "finds" capable of changing the interpretation of the whole "site". This approach also allows the specialist information needed to interpret the material in the "pits" to be presented only in the outline needed to give the bigger picture or support the main theme, but will demand further reading from other sources if more detail is required!

## Specific Objectives

As an introduction to this book it is necessary to identify who might be interested in its contents. The initial objective was to present "*the essentials of computer graphics*", in other words to enable people to create graphical products using a computer. The problem with this statement is that there are many levels at which this task can be approached in a modern computer system.

If a computer system is set up with a graphics user interface already in place, then learning to use it can be considered to be very close to learning "*the essentials of graphics*". This is because such a computer system can provide a working context similar to that in which artists, designers and draughtsmen have traditionally worked. Physically moving a "mouse" can be used to push a "pointer" around a display surface leaving marks of various kinds. Different "brushes" can be associated with the mouse, which allow areas to be painted, textured or shaded, as the artist or designer chooses. Similarly text and symbols can be created, selected and placed as desired anywhere on the display surface.

Consequently the "*essentials of computer graphics*", has been taken to mean the task of making a computer system produce displays without an interactive display system being in place between the user and the computing system. In this exercise the initial targets of the presentation are constructing the components of a system, which can provide the display environment required by an interactive graphics user interface.

What is the essential ingredient of this approach? It is that all images or drawings have to be constructed by issuing computer language commands. For an artist this amounts to producing a painting or drawing "indirectly", issuing verbal instructions to a second party while blindfold. This restricts the process to those actions that can be defined in unambiguous language statements. A very difficult task if all the sophisticated internalised knowledge and experience of the artist is to be accessed. In fact it is virtually impossible which is why a "computer-aided" environment employing an interactive graphics-user-interface (GUI) must be provided for many application that need to use feedback from this kind of human expertise.

At the risk of oversimplifying the situation, there seem to be two communities of readers for a book outlining the essentials of a computer graphics system in this way. Those who already can produce graphics manually, or within a graphics painting or drawing environment, but have not either needed to, nor had the opportunity to learn to express their actions in an explicit computer language form. And those who can program computers and therefore are accustomed to working in this way, but who have not yet explored the use of a computer language to generate pictures.

The aim is to satisfy both these groups, because once this has been done then a next stage can be moved into: *"using an interactive graphics environment"* to build customised application systems: a subject of interest to both communities. This can then be extended to address the more general theme, which can be presented as *"integrated computer and graphic modelling"*. In this context graphics becomes a primary medium for communication between users and the computing system. The emphasis is providing support for content generation in specialist application work.

For example making a system to help in the post-production work on films, but not the post-production work itself. Where the interest becomes "content generation" for a particular application this becomes the subject matter for a different book.

In conclusion the technical foundation of this subject can be summarised as the use of algebra to represent geometrical relationships in order to use computer language statements to create pictures. The overall subject of graphics as a modelling medium opens up a wide range of topics that are too extensive to cover even in a pair of books. Consequently only the main ideas are presented that are required for the graphics and spatial modelling tasks that are examined in the text.

For some of the mathematical topics a full treatment will have to be found in specialist texts. Similarly a full treatment of the programming language Java will need to be found from other sources. Java code is provided wherever possible because in general-outline, an algorithm can appear simple, but the real programming difficulties lie in the special cases and the details of implementing the algorithm as a robust working program.

What is interesting is the range of mathematics that is needed to set up useful graphic and three dimensional models within the context of computer programming is small compared with the range of mathematical results that exist, that are potentially applicable to this area of work. The starting point could hardly be simpler: write a program to generate a list of properties such as colour values for each cell in an array of pixels and allow the display hardware to present this list as an image.

Selected Java programs to support the text will be added to the website at www.springer.com/978-1-84800-178-7.