Use of computers and cathode-ray-tube displays in visual psychophysics. Part I

HANS STRASBURGER*

Institut für Medizinische Psychologie, Otto-von-Guericke Universität, D-39120 Magdeburg, Germany
Institut für Medizinische Psychologie, Ludwig-Maximilians-Universität, D-80336 München, Germany

'Welcome to the machine'
Roger Waters, Pink Floyd, 1975

Computers have become so much a part of our lives that it takes a power failure to remind us of their presence. One area in which they have become indispensable is in the methodology of visual psychophysics, which forms the topic of this special issue. A typical laboratory setup for psychophysics, two decades ago, was a Maxwellian-view system and an optical bench—perhaps linked to a PDP-11 'laboratory computer'. Today, it is a personal computer, a cathode-ray-tube (CRT) display for stimulus presentation, and a set of software applications for data analysis and presentation. However, the CRT display was not meant for psychophysical experimentation—it is *designed* to be a compromise between image quality and cost—and its precision is still surpassed by that of optical apparatus. Moreover, the flexibility of the computer, which seems fit to run any conceivable experiment or to solve any task, can be as challenging as ever for a seemingly simple experiment off the beaten path.

Hardware power, on several measures, doubles every 18 months and the number of users in the world increases at a similar rate of growth. This exponential trend has been surprisingly stable for half a century and has led to dramatic changes in the computer's role in society. Yet the usefulness of hardware and software *combined* to solve a certain task increases quite slowly, and not always steadily, slowed down through bulkier operating systems, feature-overloading driven by market forces, and death of products that are out of the current main stream. Psychophysical experimentation had been among the forerunners in applying what was called the mini-computer; today it is but a small field of computer application and the rapid technological change at the same time offers exciting, unprecedented chances and poses unexpected problems: Powerful software environments have become available that allow the design and running of whole experiments, the preparation of sophisticated stimuli, the monitoring of

---

*hans@imp.med.uni-muenchen.de

real-time variables, statistical analysis, data organisation and presentation, and mathematical modelling. The fascinating list is endless, and some of these systems and techniques will be presented and discussed here. At the same time, valuable software can become unusable when the hardware and software environment—platform, operating system, co-operating software, data format, etc.—changes too quickly, resources for porting and particularly for support are scarce and their importance is underrated.

The main obstacle to the re-use of any software or hardware component in a different context is incompatibility. Perhaps we want to use a new software-based technique of improving grey-scale resolution in a certain setup of stimulus presentation? Even though the scientific community has a strong interest in exchanging ideas and sharing solutions and data, incompatibility is still infuriatingly with us: for each compatibility problem solved, another one springs up. The main line of separation is still between platforms and operating systems but numerous other areas of incompatibility are equally important. In effect, the community is split into a large number of compatibility groups who each use a specific combination of operating systems, computer languages, applications, and data formats. The most common platforms are currently MS-DOS, Windows 3.x, and Windows 95 on the IBM and compatible (the 'PC'), the MacOS on the Macintosh, and a number of Unix windows systems—most notably the X-Windows system—on workstations. And in a year's time there will be new, and perhaps slightly incompatible, versions of each software component.

Sometimes we perceive software (or specialized hardware) as a part of science; other times we perceive it as a commodity, commercially available and to be acquired as needed. In the first case, 'software as methods', the best place for a description might be those same journals where the methods are applied; in the second case, an advertising leaflet or a colourful Web page seem more appropriate. This ambiguity in the role of software, or instrumentation in general, might underlie the emergence of specialized *methods* journals, of which excellent examples exist and can be found in the References section of each paper herein. But by confining methods to specialist publications, the community is done a disfavour: interesting techniques go by unnoticed and the wheel is re-invented once again. In this special issue, we want to foster the contact between those who have developed a technique and those who might like to apply it, so we take a middle road: In addition to a fine set of methodological papers, a section of Software and Hardware Notes provides descriptions of packages selected on the basis of their *usefulness* to the researcher. These short notes concentrate on *where* and *how* to use a particular piece of software or hardware. They can thus serve as pointers which alert the reader to a potential solution and help him or her decide whether to go further and obtain more detailed information.

THE CONTENT OF PART I

The large number of contributions has made it necessary to divide the material into two issues, the separation being largely arbitrary. Each issue contains a section of full and short papers, and a section of Software and Hardware Notes. This is the first of the two issues.

Papers section

In the papers section, several contributions focus on the CRT, its characteristics, limitations, and its calibration: Wolf and Deubel and Di Lollo *et al.* study phosphor persistence, each from a complementary viewpoint. Wolf and Deubel analyse persistence of patterns on a constant-luminance background, as in a sinewave grating, and stress the importance of physical measurements. Di Lollo *et al.* analyse persistence of patterns on a dark background and stress the importance of psychophysical measurement. Pelli warns of spatial interactions on the CRT screen that result in marked deviations of luminance from what is expected solely from the gamma nonlinearity and screen inhomogeneity. Bohnsack has performed measurements on a specific display. Bach describes a technique of CRT luminance calibration and another such technique on the same subject is contained in Tyler and McBride's note later on. Finally, on the subject, Bach, Meigen and Strasburger present an overview of CRT limitations and pitfalls, with a structuring scheme for analysing such artifacts.

There is one paper on specialized hardware and there are two on software techniques. For studying interactions between eye movements and perception, Van Diepen describes surprisingly simple hardware that makes the image content dependent on the locus of fixation. Gofen and Mackeben provide practical solutions for a common problem that recurs with every new operating system: that of stimulus timing, for the two currently predominant Windows operating systems on the PC. Finally, Tyler describes a full-colour version of his fascinating, purely software-based technique of increasing luminance resolution at the cost of an unnoticeable reduction of colour resolution.

Two empirical papers study perceptual consequences of using CRTs. CRTs are often considered as perceptually equivalent to other ways of stimulus presentation as long as certain thresholds are not exceeded, in parameters such as frame-rate, spatial resolution, and linearity, for example. It is important to note that a general equivalence under all conditions cannot be true. Montegut, Bridgeman and Sykes show a small, but systematic, influence of frame rate on reading speed. And Kulikowski, McKeefry and Robson describe the pitfalls of using CRTs to separate colour-specific from luminance-only perceptual mechanisms, which has consequences for the diagnosis of ophthalmic disorders.

Software and Hardware Notes

The section 'Software and Hardware Notes' contains short descriptions of software packages and of specialized hardware. The scope here has been widened, to include more of the specialized software that is of interest to the psychophysicist. Here in Part I, two papers describe packages that are platform independent, three predominantly for the Macintosh, five for the PC, and one for the X-Windows system. There is one paper that is predominantly on hardware.

Of the platform-independent packages, Verlinde presents software, written in C++, for generating random-dot displays. Watson and Solomon describe three Mathematica Notebooks (i.e. software libraries for the *Mathematica*™ language). Two run

on any platform that supports Mathematica; one is Mac-specific (and uses Pelli's *VideoToolbox*).

For the Macintosh, <u>Pelli</u> briefly describes his well-known and widely used, comprehensive *VideoToolbox*. <u>Brainard</u>'s *Toolbox* is an extension to the *MATLAB* language that uses Pelli's *VideoToolbox* to provide the low-level interface to the video hardware. <u>Tyler</u>'s *Morphonome* is a plug-and-play application for psychophysical experimentation on the Mac including the detection of pattern, motion, colour and noise. In particular, it allows low-contrast measurements, like obtaining a contrast-sensitivity function, for which it uses an implementation of Tyler's bit-stealing algorithm mentioned earlier.

On the PC, a comprehensive library, which is also the basis of an extensive environment for psychological experimentation, is <u>Irtel</u>'s *PXL* (pronounced 'pixel'). <u>Swift, Panish and Hippensteel</u> describe *VisionWorks*, a powerful and comprehensive, commercially available software system for visual psychophysics. Finally for the PC, three small packages are presented: <u>Jüttner and Strasburger</u> describe an interface between Fortran and Irtel's *PXL*, to allow the porting of Fortran applications to the PC without the need for rewriting. On that basis, <u>Strasburger</u>'s program for measuring contrast sensitivity for the recognition of patterns now runs on standard PCs. Finally, <u>Kasten, Strasburger and Sabel</u> present a small suite of perimetry programs for use in neuropsychological diagnosis and rehabilitation.

Unix, due to its bulkiness and design for non-real-time applications, has never been popular for the control of experiments. But for tasks that are not time-critical, the powerful X-Windows system is a desirable environment. <u>Lesher</u> presents *XPIP*, a toolkit for building graphical user interfaces under X-Windows that is suited for rapid preparation of static stimuli, such as Kanisza figures.

Finally, there is one Note on specialized hardware, together with supporting software: <u>Finley</u>'s 'point-plotting buffer' is a system that allows short stimulus presentation times and fine control of timing, down to the low-millisecond and microsecond range, which cannot be achieved with standard raster-scan CRTs. I only know of two such systems that allow the study of certain rapid visual mechanisms (this, and another one from M. Fahle at the eye clinic in Tübingen, Germany) and it is unfortunate that the required CRT technology is in danger of becoming obsolete. It needs the support of those interested to encourage small manufacturers not to eschew such niche applications.

The search for the right technique is constantly with us when we plan our experiments. I hope you will find this special issue useful, challenging, and interesting—and perhaps you will feel the excitement and joy we had in putting it together.

Use of computers and cathode-ray-tube displays in visual psychophysics. Part II

In this second part of the special issue we continue the collection of technological papers centred on the use of the CRT in vision research. As before there are two sections: first, full papers and then short descriptions of software.

PAPERS SECTION

In the papers section, four contributions are concerned with the technology of presenting dynamic stimuli, each concentrating on a different problem: Cox overviews and discusses software-based techniques for presenting movement stimuli with a computer. Mulligan discusses the application of video-compression hardware for the same purpose. Intriligator presents his real-time technique of creating movement stimuli in random-dot patterns. Szatmary, Hadani and Julesz use motion parallax in random-dot patterns to create a monocular depth cue. Next is a paper on the subject that was the main focus in Part I of the special issue, the CRT monitor itself: Shepherd provides practical help for circumventing the pitfalls in display colour calibration. Last in this section, Cornelissen *et al.* show us how to present CRT stimuli during functional magnetic-resonance imaging, that is, in an environment which precludes direct view of a CRT screen.

SOFTWARE NOTES

As in Part I, the section containing software notes covers a wide spectrum of use in visual psychophysics. A major problem is stimulus presentation, which has many interpretations. To the ophthalmologist and optometrist, visual stimuli are a tool for diagnosing visual function in the early visual pathway and their physical characteristics play an important role. Bach explains the techniques used in his *Freiburg Visual Acuity Test* that achieve adequate physical stimulus precision on standard Macintoshes. French presents the *UMIST Eye System*, a comprehensive toolbox on the PC for the practising optometrist. The visual physiologist shares the ophthalmologist's interest in the early pathway, but to him the stimulus is a means for activating nerve

cells and a requirement is that stimulus and data collection are synchronised in time. Bohnsack and Troy present *VSApc*, a package for that purpose.

To the psychologist stimulus presentation is part of a more complex series of events and an added requirement is to adjust to that complexity. MacWhinney, Cohen and Provost present *PsyScope*, a comprehensive system for psychological experimentation on the Macintosh; the general area of application is similar to that of Irtel's system for the PC presented in Part I of the special issue. Hübner's *Visuelle Welt* is for use in the classroom, for elementary courses in perception.

To the engineering scientist in computational and machine vision, visual stimuli are the result of mathematical operations, and as such they help in testing computational models of visual processes. Caelli, Dillon, Osman and Krieger present their large, general purpose system for image manipulation and machine vision *IPRS*, to which many people have contributed. Fiser and King have a specialised package for wavelet analysis of images.

To round off the scope of software for the psychophysicist, the last four papers are on specialised statistical methods. Three of these have to do with reliable determination of sensory thresholds: Harvey and Treutwein each presents their technique for rapid threshold acquisition by adaptive stimulus placement, *ML-PEST* and *YAAP*, which use different statistical methods and computer languages. Foster and Bischof's software is for estimating by bootstrapping, a Monte-Carlo resampling approach, the accuracy of thresholds thus obtained. Macmillan and Creelman's *d'plus* is for a different range of applications; it provides accuracy and bias statistics under Signal-Detection and Choice-Theory models for a number of experimental paradigms. The theory behind the program is laid out in the authors' well-known book. These statistical papers conclude Part II.

Hans Strasburger
Institut für Medizinische Psychologie
Otto-von-Guericke-Universität
D-39120 Magdeburg
Germany