PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

System Architectures for Interactive Knowledge-Based Image-Interpretation [and Discussion]

C.J. Taylor, J. Graham, D. Cooper and D. Lane

Phil. Trans. R. Soc. Lond. A 1988 **324**, 457-465
doi: 10.1098/rsta.1988.0033

Phil. Trans. R. Soc. Lond. A **324**, 457–465 (1988)
Printed in Great Britain

System architectures for interactive knowledge-based image-interpretation

By C. J. Taylor, J. Graham and D. Cooper

Department of Medical Biophysics, University of Manchester, Stopford Building, Oxford Road, Manchester M13 9PL, U.K.

[Plates 1 and 2]

We discuss hardware and software architecture for automated image-interpretation. The importance of considering the complete system is emphasized leading in particular to the conclusion that high-level and low-level processing are intimately linked. We present arguments to support the idea that automated image-interpretation systems should be knowledge-based and interactive. We attempt to identify the main architectural problems which such systems must address and outline a systematic strategy for acquiring, structuring and using knowledge.

Introduction

In this paper we consider the architectural issues raised by systems for automated image-interpretation. System architecture involves both hardware and software and we attempt to discuss their interrelation, arguing in the end that software problems are the more crucial. We begin by considering the broad characteristics of the image-interpretation task and identify high-level and low-level processing as qualitatively different activities. We review some of the specialized hardware configurations that have been developed to deal with the large quantities of data contained in images, highlighting the tendency for low-level processing to be considered in isolation. Against this background we discuss the desirability of a knowledge-based approach and suggest that interaction between the user and the system is also an important issue. We attempt to identify the central architectural problems and argue that they are not amenable to solution simply by the application of massive computing power but rather are truly architectural in nature and require a strategy for selecting, structuring and using information. Finally we propose a systematic approach to some of the fundamental problems that have been identified. We describe this as a software architecture to emphasize that architecture is not concerned solely with hardware. The architecture makes use of explicit image models and is the subject of current research.

Where possible we have used, as illustrations, practical examples from our own work in medical and industrial image-interpretation. We believe, however, that the ideas are perfectly general and have attempted to relate them to remotely sensed imagery where possible.

The first illustrative problem is chromosome analysis. Figure 1*a* shows the genetic material from a single human cell arranged, as it is during cell division, into bodies of characteristic size and shape called chromosomes. When such images are used clinically to detect genetic abnormalities, a technician uniquely identifies each of the individual chromosomes by its size, shape and pattern of stain uptake (banding pattern). They are normally arranged in a regular

display called a karyogram (figure 1*d*). A machine that could take such cells and automatically generate a karyogram would be ideal. In practice, a semi-automated, interactive system is clinically useful (Graham 1988).

The second illustrative problem is automated industrial inspection, specifically the inspection of motor-car drum-brake assemblies (figure 2). In this case the task is to recognize and locate each of the component parts (which are moveable) and check that they are present, correctly fitted and undamaged (Woods *et al.* 1987).

Levels of processing

It is convenient to identify two qualitatively different types of processing that are required for automated image-interpretation. Low-level processing is primarily numerical and acts directly on image data. Examples are geometric correction, stereo ranging, homogeneous region extraction and edge-detection. In general, low-level processing takes one or more images as input and produces a different image as output. Typically, the result at a point in the output image depends on the intensity values in a small neighbourhood surrounding the corresponding point in the input image. An example is edge-detection which is shown in figure 2*b*, *c*.

High-level processing is largely symbolic and involves recognizing and describing image structures. Typically this involves matching the observed data to some model of the expected appearance of known structures. For instance, for the brake assembly we might store an idealized edge map for each subcomponent and attempt to match these to the edges detected in an observed image. In remote sensing a comparable example might involve the use of map data as a model to which observed data must be related.

High-level and low-level processing present very different characteristics. Low-level processing typically involves very simple repetitive operations performed on large data sets, whereas high-level processing is essentially symbolic and involves complex processing on relatively small data sets. It is important to recognize that both types of processing are involved in automated image-interpretation and to reflect this in system architecture.

Specialized hardware

In this section we consider some of the hardware arrangements that have been developed to address specific problems posed by image-processing. Much of the effort has concentrated on low-level processing where significant computing power must be brought to bear, though we argue later that these problems should not be considered in isolation.

Coprocessor systems

Figure 3*a* shows the arrangement of a coprocessor system (see, for example, Taylor *et al.* 1986). In such a system a relatively small number of specialized processors co-operate to act upon image data and other types of data held in memory. Operations such as data-processing, memory-address calculation and program-flow management can be performed simultaneously by processors specializing in these activities. The coprocessors can be designed and programmed to make selected primitive operations very efficient and can thus achieve realistically high performance for low-level processing. Such an arrangement is, however, extremely flexible and can also be designed to perform high-level processing efficiently within the same structure.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

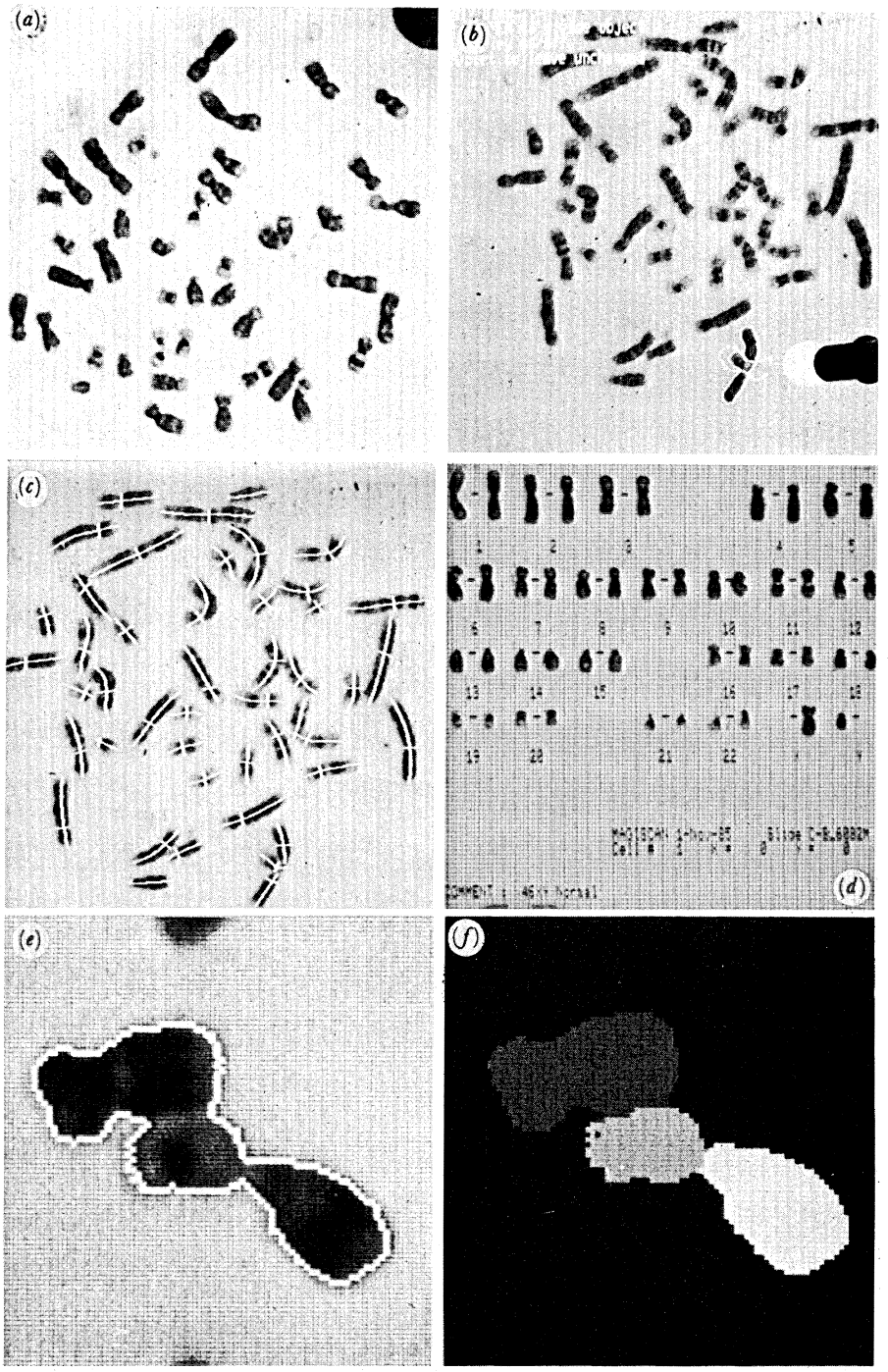


FIGURE 1. Chromosome analysis. (*a*) Microscope image of a dividing human cell; (*b*) user interacting with an automated analysis system to separate overlapping chromosomes; (*c*) axis and centromere automatically located for each chromosome; (*d*) automatically generated karyogram; (*e*) erroneous initial hypothesis for a chromosome boundary; (*f*) modified hypothesis generated as a result of obtaining further low-level evidence.

(*Facing p.* 458)

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

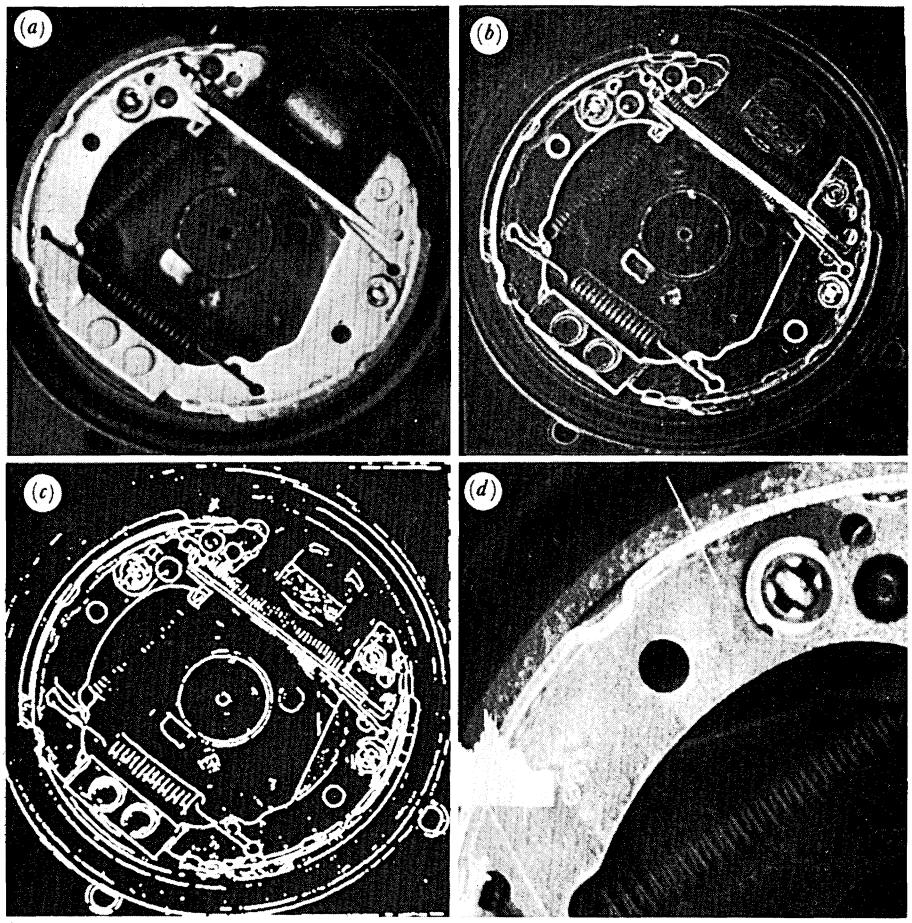


FIGURE 2. Brake inspection. (*a*) Plan view of a motor-car drum brake assembly; (*b*) edge strength image of (*a*); (*c*) detected edges from (*b*); (*d*) line which defines the position at which brake lining thickness should be measured. The intensity profile along this line is displayed and markers on the line indicate the points between which the system intends to make the measurement.

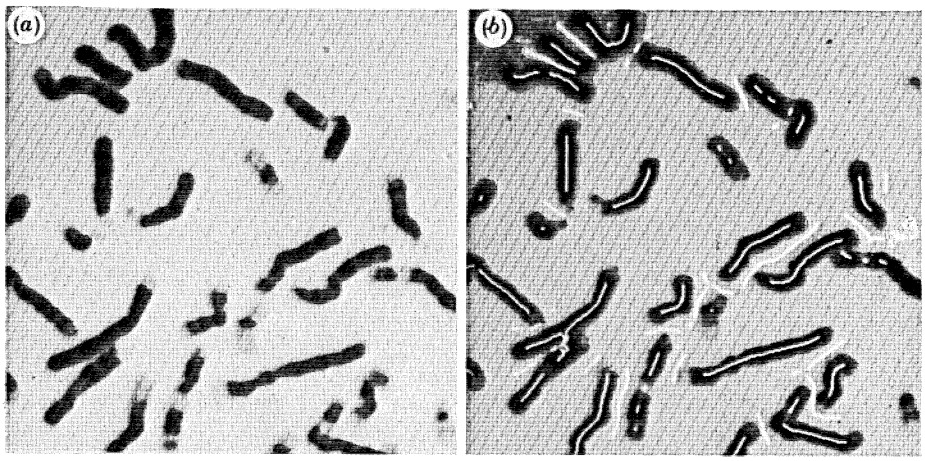


FIGURE 6. Centre of symmetry cues. (*a*) Original image-containing chromosomes; (*b*) loci of intensity symmetry detected directly from the intensity image.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Pipeline processors

A simple arrangement, which achieves high performance for low-level processing tasks, is shown in figure 3*b* (see, for example, Gerritsen & Monhemius 1981). The pipeline processor makes use of the fact that in low-level processing an identical sequence of operations must often be performed on a large number of pixel values. Each processor in the pipeline undertakes one step in the sequence of operations. Each pixel value is sent down the line in turn so that if there are n stages in the pipeline n pixels are operated on at once. If all the processors have the same speed then the system is n times as fast as an individual processor. Once high-level processing is involved the arrangement offers virtually no assistance.

Array processors

The arrangement of an array processor is shown in figure 3*c* (see, for example, Duff, 1979). These systems make use of the characteristic of many low-level algorithms that they involve operations on small neighbourhoods of pixels. In this radically different arrangement there is a processor and memory element for each pixel. Each processor–memory element is connected to each of its neighbours so that it can act upon its own and neighbouring pixel values. Thus by sending the same instructions to each processor an operation such as edge-detection can be performed for each pixel in the image simultaneously. Again, although this configuration is well suited to low-level processing it is inappropriate for high-level processing because the connections between processors are insufficiently general.

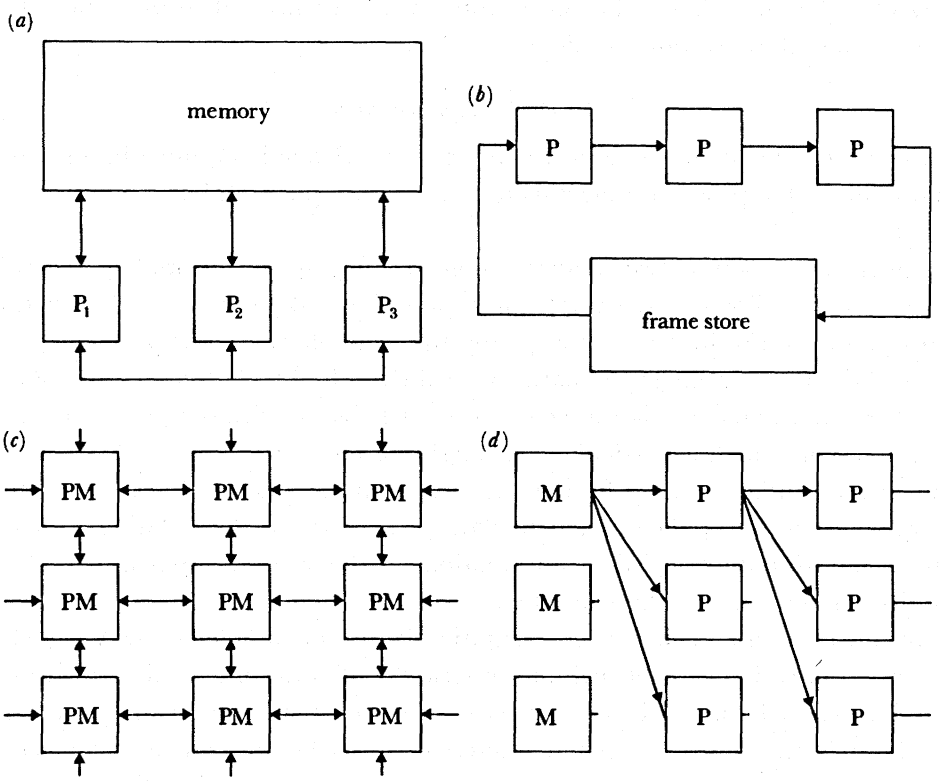


FIGURE 3. Specialized hardware for image processing. (*a*) Coprocessor system; (*b*) pipeline processor; (*c*) array processor; (*d*) adaptive network. P, processor; P_t, specialized processor; M, memory.

Adaptive networks

Adaptive networks (figure 3*d*) bear a superficial resemblance to the organization of the brain and represent a more radical architectural approach (see, for example, Hopfield 1982). Here the processors only ever execute one type of operation, combining the set of inputs according to a predetermined rule to produce an output. The nature of the processing that takes place is thus determined not by the processors but by the pattern and strength of the connections between processors. In principle, the final output of such a system can be either another image or a symbolic interpretation. Adaptive networks are very interesting but so far no-one knows much about how to set up the connections to achieve the desired result or indeed about their generality. A particular concern is that knowledge of the problem domain must necessarily be incorporated implicitly into such systems, a characteristic which, as we argue in the next section, is undesirable.

Interactive knowledge-based interpretation

Images from a particular domain, such as remote sensing, are normally interpreted by making use of specialized knowledge. This knowledge includes the nature of the interpretation task together with the identity and expected appearance of key objects and structures. Automated systems have been developed to interpret complex images (Aleksander 1983; Brunt *et al.* 1983; Dixon & Taylor 1979; Graham 1987; Pycock & Taylor 1980; Thomason 1986; Tucker & Shippey 1983; Woods *et al.* 1987) but generally the prior knowledge involved has not been easily identifiable; it has rather been implicit and embedded in particular algorithmic approaches to the interpretation problem.

Knowledge representation

A knowledge-based system is one in which prior knowledge is supplied in an explicit form, quite separate from the program that puts it to use. This approach has a number of potential advantages:

(i) the clear separation between application-specific knowledge and a general-purpose interpretation engine allows more complex interpretation tasks to be tackled reducing the cost and engineering expertise involved in applying the technology to new problems;

(ii) knowledge can be applied systematically as the result of an automated reasoning process whereas systems that make implicit use of prior knowledge require the programmer to foresee the circumstances in which a particular fact might be relevant;

(iii) knowledge can be acquired from a number of sources and used in a coherent manner; thus in a remote-sensing application, map data, expert knowledge relating to feasible configurations of land-use and statistical information on the size and appearance of regions of known land-use might be usefully combined.

An important issue in knowledge representation is that of completeness. A model is a form of representation in which geometrical and intensity configurations are described in sufficient detail that images of feasible objects and structures (or their significant features) may be generated. Models are thus of particular interest because they provide a means by which an automated image interpretation system may arrive at a complete explanation of each observed image (Ayache & Faugeras 1986; Brooks 1981; Hanson & Riseman 1978; Pollard *et al.* 1987).

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

In addition to a method of knowledge representation a practical system requires a means by which the user can supply relevant knowledge. Although some knowledge can be presented to the system formally (in a manner analogous to conventional programming) there is also a need for interactive dialogue between the user and the machine. Two types of interaction may be considered. The fact that both involve the user gaining rather direct access to the geometrical and intensity structure of an image provides additional support for a model-based knowledge representation that can form a natural link between the user interface and the internal-processing régime.

Offline interaction

Although some knowledge can be stated simply, a great deal of what is used to interpret images can only be conveyed conveniently by showing examples. This is indeed the method we often use when explaining a complex visual task to another person. The requirement is that the user can present example images and interact with the system in such a way that it extracts salient features. We show this with an example from the interpretation of brake assembly images.

In figure 2*d* the system has identified the position at which the thickness of the brake lining is to be measured. The intensity profile observed along the measuring line does not, however, allow the inner and outer edges of the lining to be easily identified. The figure shows the system making an educated guess at the location of the inner and outer boundaries. The user accepts the guess if it is accurate or corrects it with a lightpen if it is erroneous. This is repeated for a number of examples and because, in each case, the system has both the observed intensity profile and the correct interpretation it can, by building an internal model, learn how to guess correctly so that no interaction is required at run-time.

Online interaction

In many circumstances it may be desirable for the user to provide additional knowledge at run-time. This may be appropriate because complete automation is too difficult or because a user-guided system is more appropriate for the task in hand. We show this with an example from chromosome analysis (figure 1). Here the generation of a karyogram can be significantly automated but occasionally configurations arise that are difficult for the system to resolve. In any case, because the system is used diagnostically, skilled supervision is desirable. The system starts by determining a boundary for each chromosome. Figure 1*b* shows how the user can separate overlapping chromosomes with the lightpen. To decide which chromosome is which, an axis of symmetry must be obtained for each and used to define a path along which the banding pattern should be measured and the position of the centromere located. Figure 1*c* shows a display of the results, which can again be corrected with the lightpen. Finally the chromosomes are classified and displayed in a karyogram, which can be modified if necessary by moving chromosomes with the lightpen (figure 1*d*).

The architectural problem

Having identified some of the important characteristics of an ideal image-interpretation system we can consider some of the architectural issues which are raised. First let us look at the question of computational complexity. It is often argued (Sternberg 1980) that this is a particularly important issue in low-level processing because of the large quantities of data

involved. Figure 4 shows with a simple example that there is also a problem with high-level processing. The figure shows a line-drawing model of a house, composed of straight-line segments and an example image from which we may assume all the line segments can be perfectly recovered. To recognize the house we must match the model and the example line by line. The number of ways to do this goes as the factorial of the number of line segments and in this example there are approximately 10^{20} ways of matching. To recognize the house in one second by using brute force would require computing power of at least 10^{14} Mips (million instructions per second). To put this in perspective NASA's massively parallel processor provides approximately 10^4 Mips (Batcher 1980). Thus it is unrealistic to assume that the application of parallelism will solve the problem, neither can it be avoided because any of the matches might be the best one. What is required is a framework within which the system can focus its attention on likely interpretations avoiding computational effort being wasted on those which are unlikely.

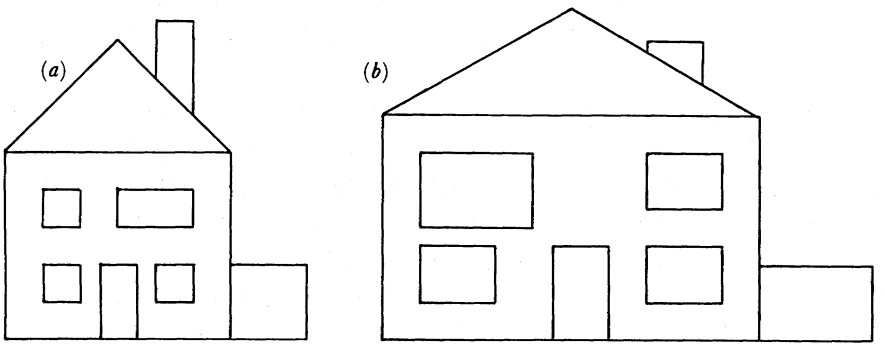


FIGURE 4. (*a*) Idealized house model composed of straight line segments; (*b*) the line segments which might be obtained for an example of a house image.

To appreciate fully the nature of the architectural problem with which we are faced we must address a further complication. In our discussion of model matching we have assumed that low-level processing may be used to extract evidence of structure (e.g. line segments) which may subsequently be interpreted by matching to a model. In practice this is unrealistic because the evidence of structure obtained by unguided low-level processing is subject to errors and will not in general represent the best evidence that could be sought in support of an emerging interpretation hypothesis. A more realistic approach is to undertake some low-level processing to provide sufficient evidence to make an initial interpretation which can then be used as an organizing hypothesis to guide the collection and interpretation of further low-level evidence. This further evidence may lead to the rejection, modification or refinement of the original hypothesis. Figure 1*e*, *f* show an example in chromosome analysis where initial low-level processing identifies a composite object. From its shape this is recognized as unlikely to be a single chromosome and further low-level evidence is sought which generates a modified hypothesis of three objects (Graham *et al.* 1986).

In summary there are two main conclusions which we draw from this discussion:

(i) high-level and low-level processing are intimately linked and the interface between them is crucial;

(ii) real visual tasks involve massive computational complexity and the central issue is that of deploying the available computing resources to best effect rather than increasing them.

We (and others) have designed coprocessor hardware which optimizes the interface between different levels of processing (Taylor *et al.* 1986; Graham *et al.* 1986) and have investigated strategies for focusing computing resources in a number of problem domains (Dixon & Taylor 1979; Brunt *et al.* 1983; Pycock & Taylor 1980; Woods *et al.* 1987; Graham 1988).

A SOFTWARE ARCHITECTURE

In previous sections we have argued that the problem of image-interpretation should be posed as that of explaining an observed image in terms of an explicit image model. The model will describe the objects and structures that may appear in the image and the relations between them. It will be parametrized so that a particular set of parameter values uniquely defines the geometric and (ideally) intensity configuration of a particular feasible image. Typical parameter values and measures of variability will have been obtained by offline interaction with example images. Image-interpretation involves a search for that set of model parameters that are consistent with those observed in example images and that define an image most similar to the observed image.

Hierarchical models

We showed in the previous section that the cost of establishing a correspondence between model elements and image structures rises exponentially with the number of model elements. To model visual worlds of realistic complexity requires that the matching problem be divided into a number of subproblems of manageable cost. This can be achieved in a natural way by organizing the model as a hierarchy. Figure 5 shows the manner in which the house model of figure 4 might be broken down into a number of submodels. In this example a partition of the model such as BODY will contain information describing the expected geometrical relations between submodels such as WINDOW and WALLS. Each submodel at the lowest level describes the relations between the line segments of which it is composed. If we can match each submodel independently the number of segment to segment matches which must be considered is reduced from approximately 10^{20} to approximately 10^{3}.

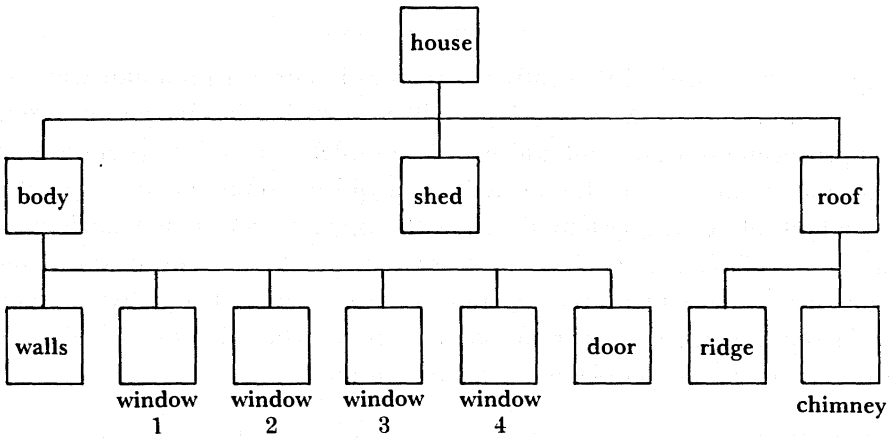


FIGURE 5. The house model of figure 4 partitioned into a hierarchy of submodels.

Cues and control

A cue is a feature that suggests that a particular model element should be instantiated. Given a hierarchially organized model and an observed image it is generally necessary to apply low-level processing to generate cues that provide evidence with which to initiate the matching process. In the house example we might detect intensity discontinuities (edges) in the observed image and starting from the bottom work upwards. Once a submodel has been matched its relation to other submodels can be used to limit the search space. For example, once WALLS has been recognized the approximate position of each WINDOW is defined and may be used as a cue which propagates the matching process.

It is important to recognize that it is not necessary to start matching at the lowest level of the model. In the house example we might, for instance, use a cue-generator that found dark blobs of about the size of a DOOR. On finding such a cue the system would try to recognize and locate the rectangle of the DOOR perhaps seeking edge information in the vicinity of the cue. If the DOOR were located the system would move up a level in the hierarchy and use the known relation between DOOR and WALLS to generate a WALLS cue. If the WALLS were successfully located then the WINDOWS could be cued. This process of cue-driven matching can continue until a complete match has been established. An important feature of this organization is that it is possible for the system both to infer higher-level models from details and details from higher-level models. Ultimately, however, the interpretation must be supported by direct evidence. It is also important to note that the pattern of control is determined, as seems appropriate, by the nature of the images to be interpreted.

Figure 6, plate 2 shows a practical example of a cue generator which operates robustly at a higher level than edges. The figure shows a chromosome image from which loci of intensity symmetry have been extracted directly without first detecting edges or separating objects from background. These lines represent candidate chromosome axes which can be refined and tested by appealing to other levels of a model to recognize chromosome boundaries, banding pattern and so on. Cues such as this which detect reasonably high levels of organization in the image are important because they will, as a result, tend to be less ambiguous.

Conclusions

In the paper we have argued that software is more important than hardware in automated image-interpretation systems. Most specialized hardware that has been built provides support for low-level processing but does not address the complete image-interpretation problem. The real issues are the manner in which knowledge is acquired, represented and used. The need is to develop methods of dealing systematically with application-specific knowledge. We suggest that explicit models of image structure offer a good means of representing knowledge internally and allow the user to interact with the system in a powerful way. The software architecture that we have outlined is the subject of current research and is believed to offer the basis of a solution to some of the architectural problems we have identified.

We thank P. W. Woods for providing pictures of the brake inspection application, and S. A. Thornham and P. J. Azzopardi for their assistance in preparing other figures.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

References

Aleksander, I. 1983 *Patt. Recog. Lett.* **1**, 375–384.
Ayache, N. & Faugeras, O. D. 1986 *IEEE Trans.* **PAMI 8**, 44–54.
Batcher, K. 1980 *IEEE Trans Comput.* **29**, 836–840.
Brooks, R. A. 1981 *Artif. Intell.* **17**, 285–348.
Brunt, J. N. H., Taylor, C. J. & Dixon, R. N. 1983 In *Physical techniques in cardiological imaging* (ed. M. D. Short), pp. 153–162. Bristol: Hilger.
Dixon, R. N. & Taylor, C. J. 1979 In *Machine aided image analysis, 1978 (IOP Conference Series no.* 44) (ed. W. E. Gardner), pp. 178–185.
Duff, M. J. B. 1979 In *Advances in digital image processing* (ed. P. Stucki), pp. 265–276. New York: Plenum Press.
Gerritsen, F. A. & Monhemius, R. D. 1981 In *Languages and architectures for image processing* (ed. M. J. B. Duff & S. Levialdi), pp. 189–203. London: Academic Press.
Graham, J. 1988 *Analyt. Quant. Cytol. Histol.* (In the press.)
Graham, J., Taylor, C. J., Cooper, D. H. & Dixon, R. N. 1986 *Patt. Recog. Lett.* **4**, 325–333.
Hanson, A. R. & Riseman, E. M. 1978 In *Computer vision systems* (ed. A. R. Hanson & E. M. Riseman), pp. 303–333. Orlando, Florida: Academic Press.
Hopfield, J. J. 1982 *Proc. Natn. Acad. Sci. U.S.A.* **79**, 2554–2558.
Pollard, S. B., Porrill, J., Mayhew, J. E. W. & Frisby, J. P. 1987 *Image Vision Comput.* **5**, 73–78.
Pycock, D. & Taylor, C. J. 1980 *Analyt. Quant. Cytol.* **2**, 195–202.
Sternberg, S. R. 1980 In *Real-time medical image processing* (ed. M. Onoe, K. Preston & A. Rosenfeld), pp. 11–22. New York: Plenum.
Taylor, C. J., Dixon, R. N., Gregory, P. J. & Graham, J. 1986 In *Intermediate-level image processing* (ed. M. J. B. Duff), pp. 19–34. London: Academic Press.
Thomason, R. L. 1986 In *Vision Conference Proceedings, Detroit, 1986*, pp. 5:51–5:61.
Tucker, J. H. & Shippey, G. 1983 *Analyt. Quant. Cytol.* **5**, 129–137.
Woods, P. W., Taylor, C. J., Cooper, D. H. & Dixon, R. N. 1987 *Patt. Recog. Lett.* **5**, 11–17.

Discussion

D. Lane (*Intelligent Automation Laboratory, Department of Electrical and Electronic Engineering, Heriot-Watt University, Edinburgh, U.K.*). In Dr Taylor's presentation he mentioned the subject of feedback, and described the way that high- and low-level processes may interact. Feedback and the ensuing issue of system stability have been much studied by mathematicians and control engineers for a number of years. Mathematical tools (pole-zero diagrams, nyquist plots, bode diagrams) have been developed to enable a designer to predict the stability margins of a system. Is the stability issue relevant in the context of knowledge-based image-interpretation, and if so, how may it be approached?

C. J. Taylor. The issue of stability is relevant, but the system with which we are dealing is much more complicated than those for which the mathematical tools mentioned were developed. Feedback is involved in knowledge-based image interpretation in the sense that high-level hypotheses may be used to guide the search for low-level supporting evidence which in turn may be used to modify the high-level hypotheses; a high-level interpretation is stable but it is not necessarily so that such a system will converge to that interpretation. The interaction between high-level and low-level processing is, however, sufficiently complex that it is difficult to see how the methods of control engineering can easily be applied.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

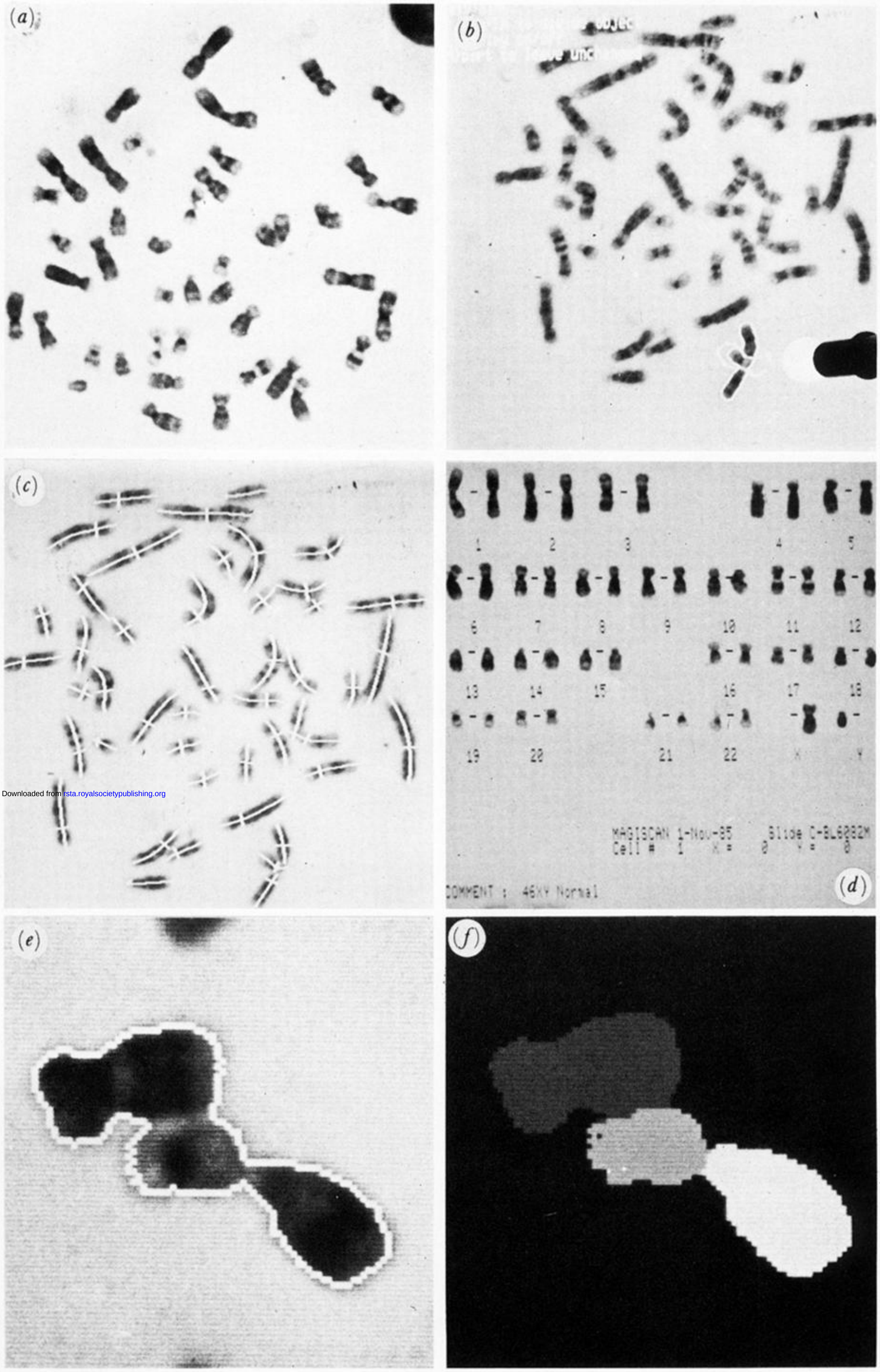


FIGURE 1. Chromosome analysis. (*a*) Microscope image of a dividing human cell; (*b*) user interacting with an automated analysis system to separate overlapping chromosomes; (*c*) axis and centromere automatically located for each chromosome; (*d*) automatically generated karyogram; (*e*) erroneous initial hypothesis for a chromosome boundary; (*f*) modified hypothesis generated as a result of obtaining further low-level evidence.

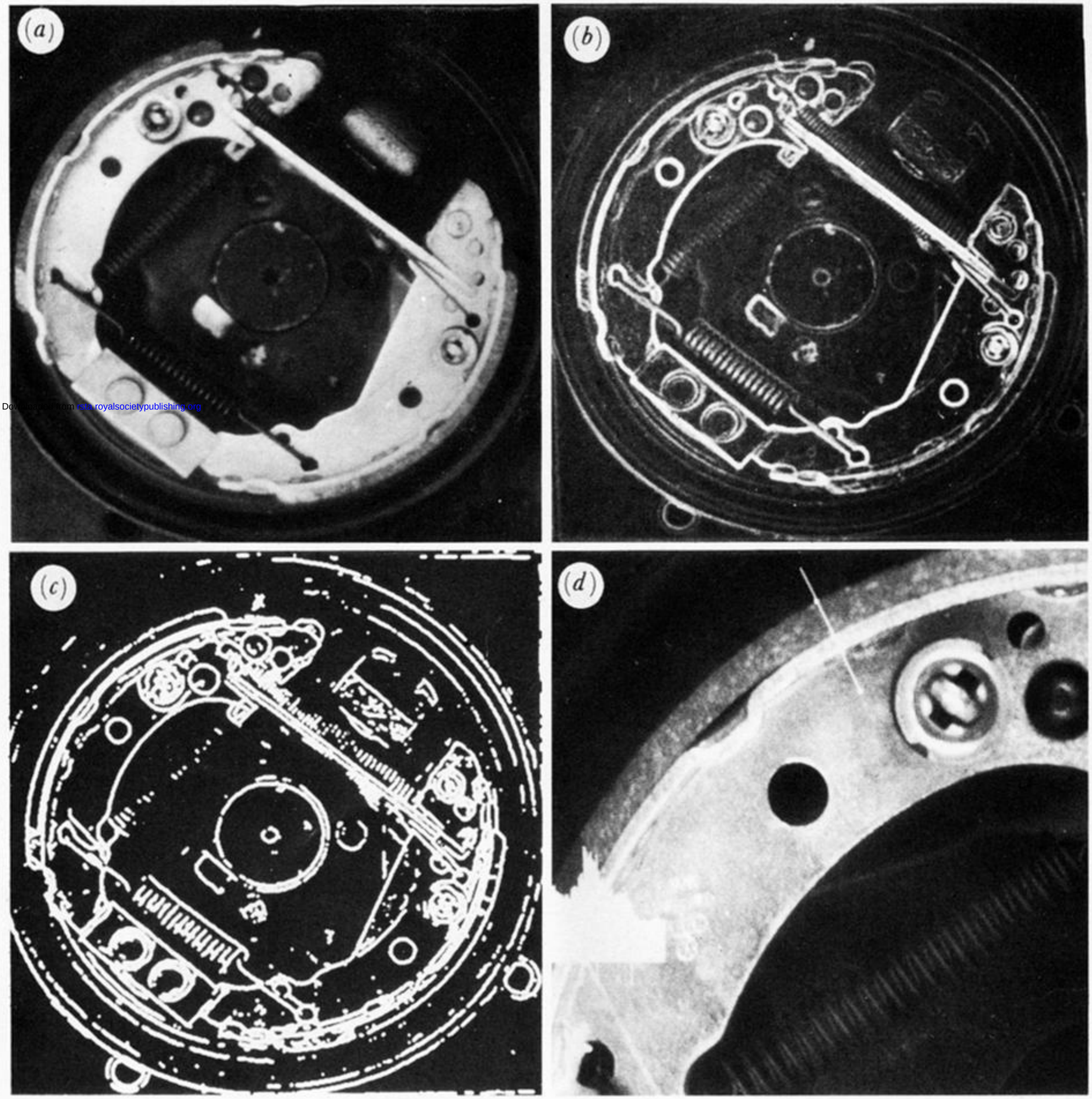
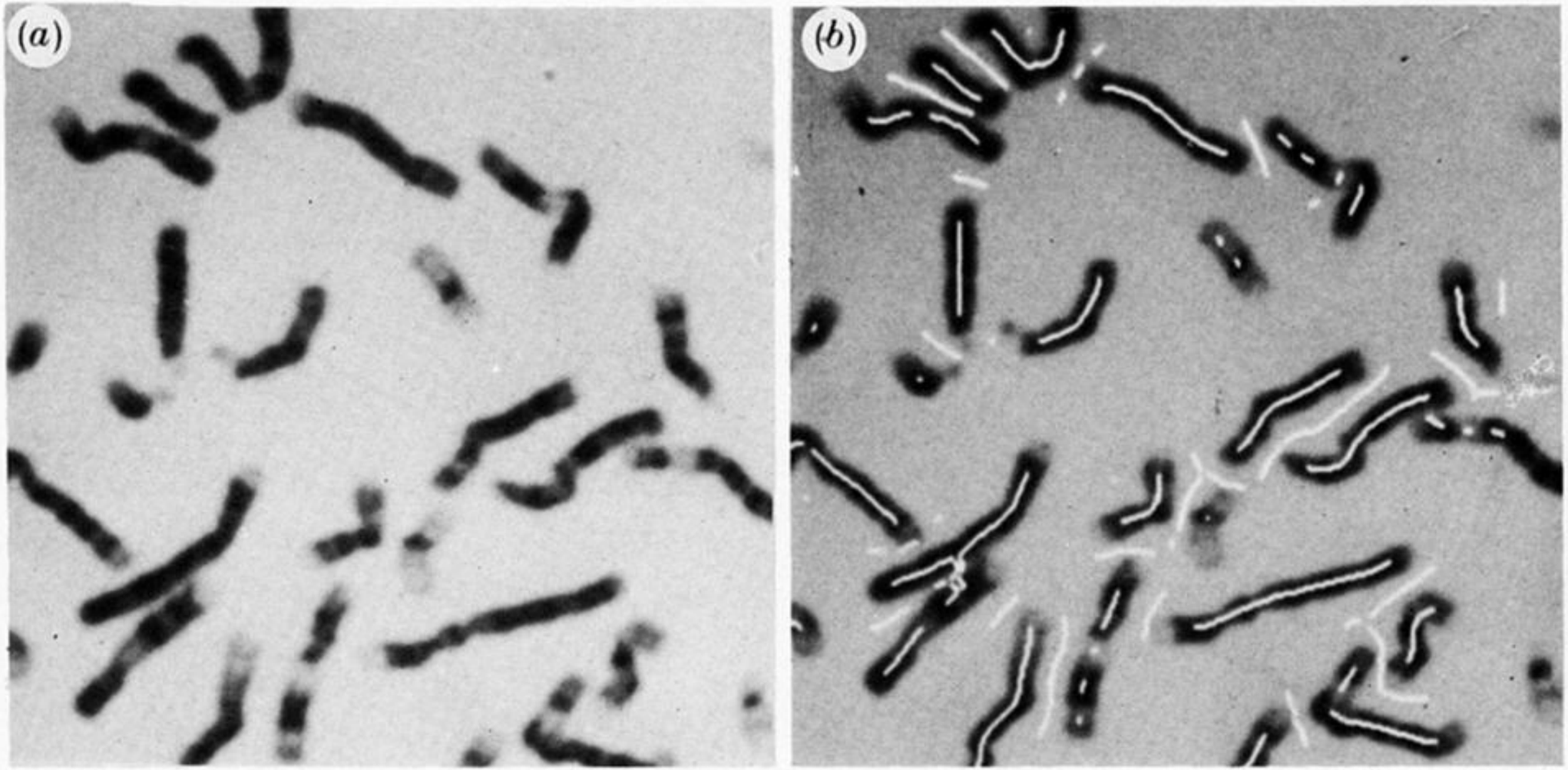


FIGURE 2. Brake inspection. (*a*) Plan view of a motor-car drum brake assembly; (*b*) edge strength image of (*a*); (*c*) detected edges from (*b*); (*d*) line which defines the position at which brake lining thickness should be measured. The intensity profile along this line is displayed and markers on the line indicate the points between which the system intends to make the measurement.



GURE 6. Centre of symmetry cues. (*a*) Original image-containing chromosomes; (*b*) loci of intensity symmetry detected directly from the intensity image.