REPRESENTING IMAGES FOR COMPUTER VISION

David Young

December 1987

Cognitive Science Research Paper

Serial No. CSRP 96

School of Cognitive Sciences,
The University of Sussex,
Brighton BN1 9QN

REPRESENTING IMAGES FOR COMPUTER VISION

David Young

Computer vision is firmly tied to one representation for images: an array of samples measured on a rectangular grid. For some work, this restriction matters little; but for much low-level processing, the representation is crucial in determining the operations that can readily be carried out. I suggest that the exploration of alternatives may be worth some effort, and to support this I investigate one possibility: the logarithmically sampled image. This representation has interesting properties and can support a number of useful operations. I show in particular how straight line detection and a fairly general form of two-dimensional matching may be performed using it, and I suggest ways in which the work may be developed.

Conventional sampling

Vision involves far more than images, but for computer vision, operations on image representations provide a vital early stage of processing. In computer vision and image processing (terms which I shall use with distinct senses below), images are converted to a representation which can be stored in computer memory by measuring the mean brightnesses of patches of the image, and representing the results in digital form. Almost invariably, these samples are taken on a rectangular grid over the image, whose layout can be represented in computer memory using a two-dimensional array. This is a natural and convenient way of representing images. In particular, the sampling is uniform: any part of the image is represented in as much or as little detail as any other part. I shall refer to this rectangular grid representation as the *conventionally sampled image*, or CSI (Figure 1).

In image processing systems, where the aim is to produce enhanced pictures for a person to look at. one part of the original image is as important as any other part; uniform sampling is needed, because it is not possible to predict where in the output viewers will direct their gaze or attention. Computer vision, on the other hand, has the goal of interpreting visual information in terms of the layout of surfaces, but it has inherited the hardware of image processing, specifically cameras and digitisers. It is also valuable in computer vision research to be able to look at the results of early processing displayed as pictures. Unfortunately, this is akin to doing image processing, and it is

tempting to forget that there is no reason why any representation used in computer vision should make sense to us when we choose to display it on a screen and allow our eyes to wander over it. Combined, the inheritance of the standard technology for image processing, and the desire to look at the results of early processing, have made conventional sampling universal as the starting point for computer vision, with the exception of a few systems using other uniform sampling schemes.

Why bother with alternatives?

In many ways, the representation of the image does not matter. If we are interested in understanding the geometrical relationships that allow us to extract surface layout from, say, linear perspective or optic flow, then we will not much care how the system holds its image data. And if the density of sampling in a CSI is sufficiently great that all the salient detail is captured, then we might argue that any operation we apply to another representation could be transformed into an equivalent operation on the CSI.

The response to the first point is simple. The application of techniques such as stereopsis, motion analysis and texture analysis depends on the development of effective image-level processes to support them. The answer to the second point lies in the fact that resources are limited: we cannot necessarily sample the image everywhere as densely as we choose; in addition an operation that is efficient in one representation may be unreasonable in another. Computer vision may be simply more effective if its early stages are based on representations that have been tailored to its requirements. Furthermore, the all-pervasive nature of the CSI may influence the way that we think about the possibilities for early processing. Specifically, operations that are uniform across the image are very easily and economically carried out on the CSI, and so low-level computer vision research concentrates on such uniform operations. It is generally assumed that any focusing of computational resources on particular parts of the image will occur only after the preliminary extraction of edges or other features has taken place.

On these grounds, it seems reasonable to suggest that alternative representations might lead to a useful extension of our range of techniques. Ideally, I should like to stand back from the problem, and try to find criteria for the adequacy of image representations, based both on a better understanding of the structure of the information available in the optic array, and on a more precise prescription of the needs of later stages of a vision system. This goal is ambitious, and all I can do here is second-best: to try to shake the dominance of the CSI by showing one alternative of some interest, both theoretically and in practice.

The logarithmically sampled image

There is a vast range of sampling patterns, but one type in particular seems worth investigation: that in which the samples lie on concentric rings, with the density of sampling rising to a peak at the centre. The particular pattern I am interested in has the same number of samples on each ring, and the pattern of sample points in any small

region of the image looks the same as the pattern in any other small region except for a rotation and an expansion or contraction (provided neither region covers the centre of the pattern). These conditions define the *logarithmically sampled image* or LSI (Figure 1), and have been chosen to provide interesting properties that can support some powerful operations. Funt (1980) has used these properties to good effect in producing a diagrammatic reasoning system based on the LSI.

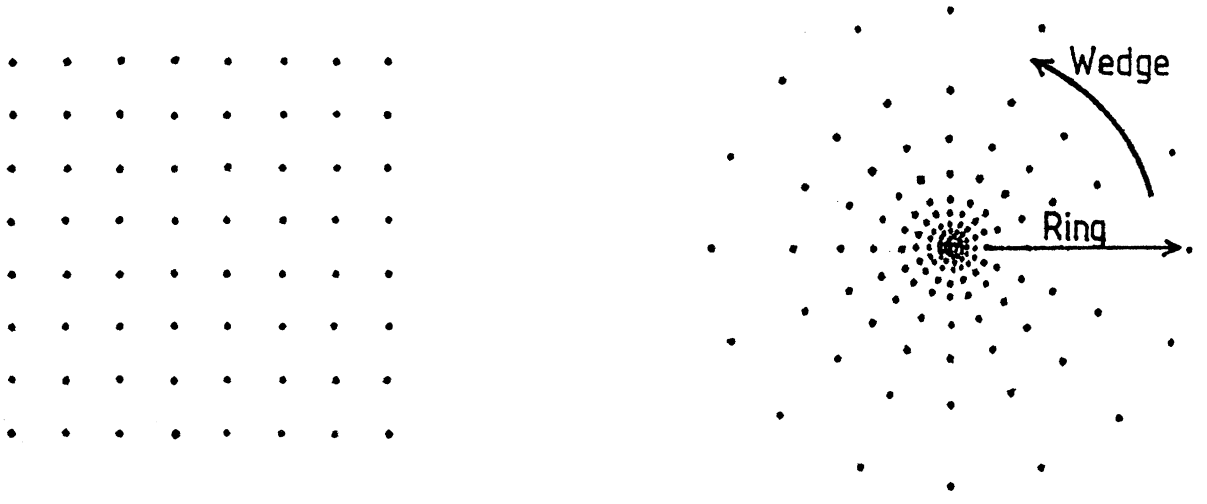

FIGURE 1: The sampling patterns of the CSI (left) and LSI (right).

A sample in the LSI may be identified by its ring number and wedge number, just as a sample in the CSI is identified by row and column indices. The very centre of the LSI can never be used, since the density of sampling increases without limit as it is approached, so a small hole has to be left there or uniformity imposed in a small region. The relation between the number of a ring and its radius is that the former is linearly related to the logarithm of the latter; hence the name. (In fact there is a more compelling reason for this name, which arises if the image surface is regarded as the complex plane, but I will not dwell on that.)

The density with which the LSI samples the image falls off very rapidly with distance from its centre; far more rapidly, for instance, than the rate at which the density of cones in the human eye falls off with distance from the foveal centre (Østerberg 1935). The fall-off means that the position of the sampling centre in the image is of great importance:

any system using an LSI must have the means to determine where to centre it at any moment in order to extract the most salient information from the image. Resources are not deployed uniformly across the image, but rather are concentrated on the part where resolution of fine detail is needed most urgently. Some strategy is therefore needed to resample the image from different centres as processing continues. The lowest level of a system using this, or any other, non-uniform sampling must become an active process.

This applies to any sample pattern with a peak of density, but the interesting properties of the LSI arise because rotations, expansions and diminutions of the image about the centre of sampling can be represented very simply. These correspond to mere uniform changes in wedge number or ring number, in the same way that in the CSI translations of the image correspond to changes in row and column. In the LSI, expanding the image involves moving every sample value outwards a fixed number of rings, and rotating the image involves moving every sample value clockwise or anticlockwise a fixed number of wedges. This gain in one kind of simplicity is at the expense of another, since in the LSI a translation no longer has the straightforward representation it has in the CSI, and LSI representations of images shifted relative to one another will no longer have the straightforward correspondence that their CSI representations have. Again, this demands that the LSI pattern can be repositioned rapidly on the image, so that we can use a resampling strategy to try to replace the inherent ability of the CSI to cope with position shifts. If the image of an object moves, the LSI will have to move with it. The techniques that are needed are, in effect, analogous to eye-movement control in human and animal vision.

The examples that follow are an attempt to show the potential value of the rotation and expansion symmetries of the LSI. I do not claim that the methods here find lines or match similar images better than existing methods; rather that lines can be found and images matched in rather straightforward ways. This seems enough to merit further work.

Finding straight lines using the LSI

Perhaps a little strangely, the LSI, with its circular pattern, is well suited to the detection of straight lines in an image. The reason is that any straight line may be mapped onto any other straight line by a combination of rotation and expansion about the centre of the LSI, regardless of where that centre happens to be (Figure 2). The LSI allows us to search efficiently through all possible combinations of expansion and rotation to discover where a straight line can best be fitted to the image.

To do this, both the image and a template with a straight line on it are represented using the LSI. It is then possible to use convolution, one of the standard methods of image manipulation, to find the amounts by which the template must be shifted in ring number and wedge number to match the image best. The output of the convolution is another LSI, but one in which the positions of peaks and troughs can be interpreted as the parameters of straight lines in the original image. A few details of my implementation are given in the appendix, and a full account will be published elsewhere, but a genuinely typical result for

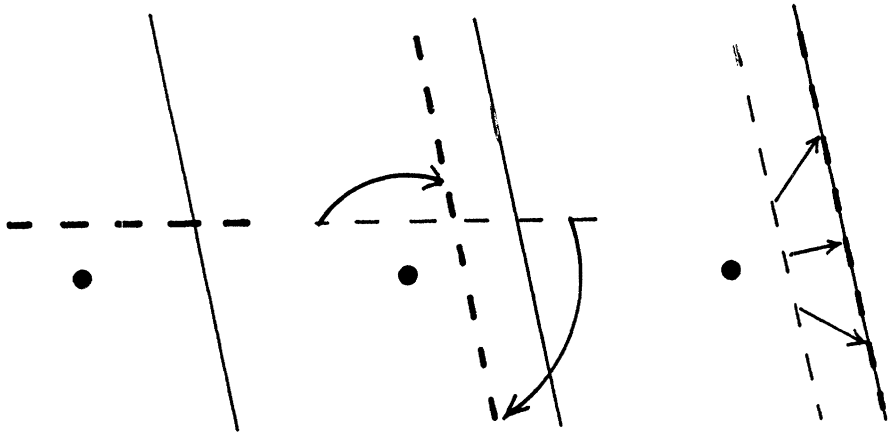

FIGURE 2: Mapping one straight line onto another with a rotation and an expansion.

an image with a good selection of straight lines in it is shown in Figure 3. The superimposed lines are the most prominent lines as determined using an LSI centred on the black dot; the method works well, at least as far as subjective assessment of the results goes, though of course in the long term that is insufficient. Quite short or weak straight lines passing close to the centre of the LSI are picked out, whereas lines further out must be either longer or of greater contrast to be detected.

Fairly general matching using the LSI

By fairly general matching, I mean finding the rotation, scaling factor (ie expansion or contraction) and translation that match one image as closely as possible to another. The ability to find such matches in two dimensions can help to support many kinds of higher-level interpretative processes. (A more general two-dimensional match would also allow anisotropic expansion, or shear, to be included.)

The CSI is good for finding the best translation match between two images that already match in scale and orientation. The LSI is good for finding the scale and orientation match between two images that already match in position, in the sense of having

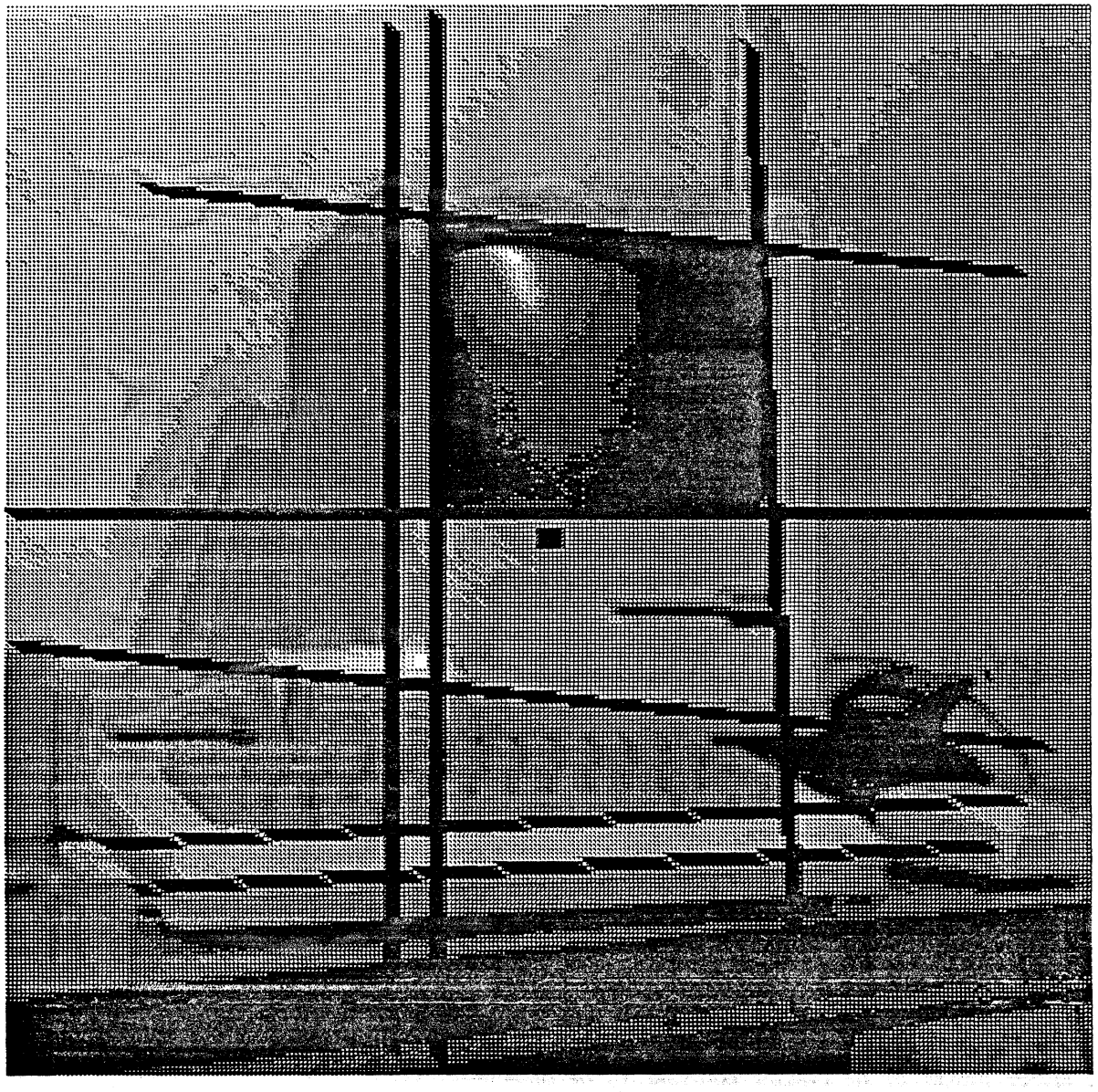

FIGURE 3: The results of straight line detection using the LSI. For more details, see appendix.

corresponding points at the sampling centres. My approach to combining the two is to use the CSI to generate a position-independent description of both images, then to use the LSI to match these descriptions in scale and orientation. This is frankly a rather opportunistic alliance of techniques, but the results are promising enough to make the approach interesting. The translation-independent description I use is the discrete autocorrelation, which may be represented in an array just as a CSI can be. The autocorrelation is a description purely in terms of offsets in the image, and it is unaffected by changes in absolute position. A great deal of information is lost when it is generated, but enough is retained to allow scale and rotation matching to be based on it. The method, then, involves correlating two LSI representations of the ordinary autocorrelations of the images to find the best scale and orientation match, and then using the CSIs to find the best position match.

Again, a few details of the implementation are given in the appendix, but Figure 4 shows a typical result. The top row shows the two original images; the one on the right was manipulated to match the one on the left. The bottom half shows the results, with on the left one of the original images transformed to reflect the results of scale and orientation matching, and on the right the results of the full match. The process is generally robust, though it is best suited to matching fairly simple shapes.

Further work on the LSI

There are some developments that can probably be carried out quickly. In the line finding technique, the lines which are found could be clipped to line segments to bring the results closer to a potentially useful representation. In fairly general matching, it may be worth exploring alternatives to the autocorrelation as a translation-independent description. Both line finding and matching need to be tried out more systematically, and more objective measures of their success obtained.

More interesting are the extensions which stem from the need to supply a resampling strategy, analogous to eye-movements in biological vision. In the line-finding example, for instance, it might make sense to resample with the pattern centre lying on one of the lines, or at the junction of some lines. Indeed, many resamplings with a rather sparse LSI may be more effective than few resamplings with a dense one. These questions need much work.

If the analysis of a static image requires strategies rather like saccadic eye-movements, then the analysis of an image sequence will require something analogous to tracking eye-movements. It goes almost without saying that the LSI is extremely well suited to picking up rates of expansion in image sequences, and hence to supplying information directly applicable to the control of action (Lee and Young, 1985). This will only work, however, if the centres of successive LSIs correspond to the same feature of the image. The LSI must track the image of a moving object, and there is a considerable challenge in finding effective ways of carrying this out.

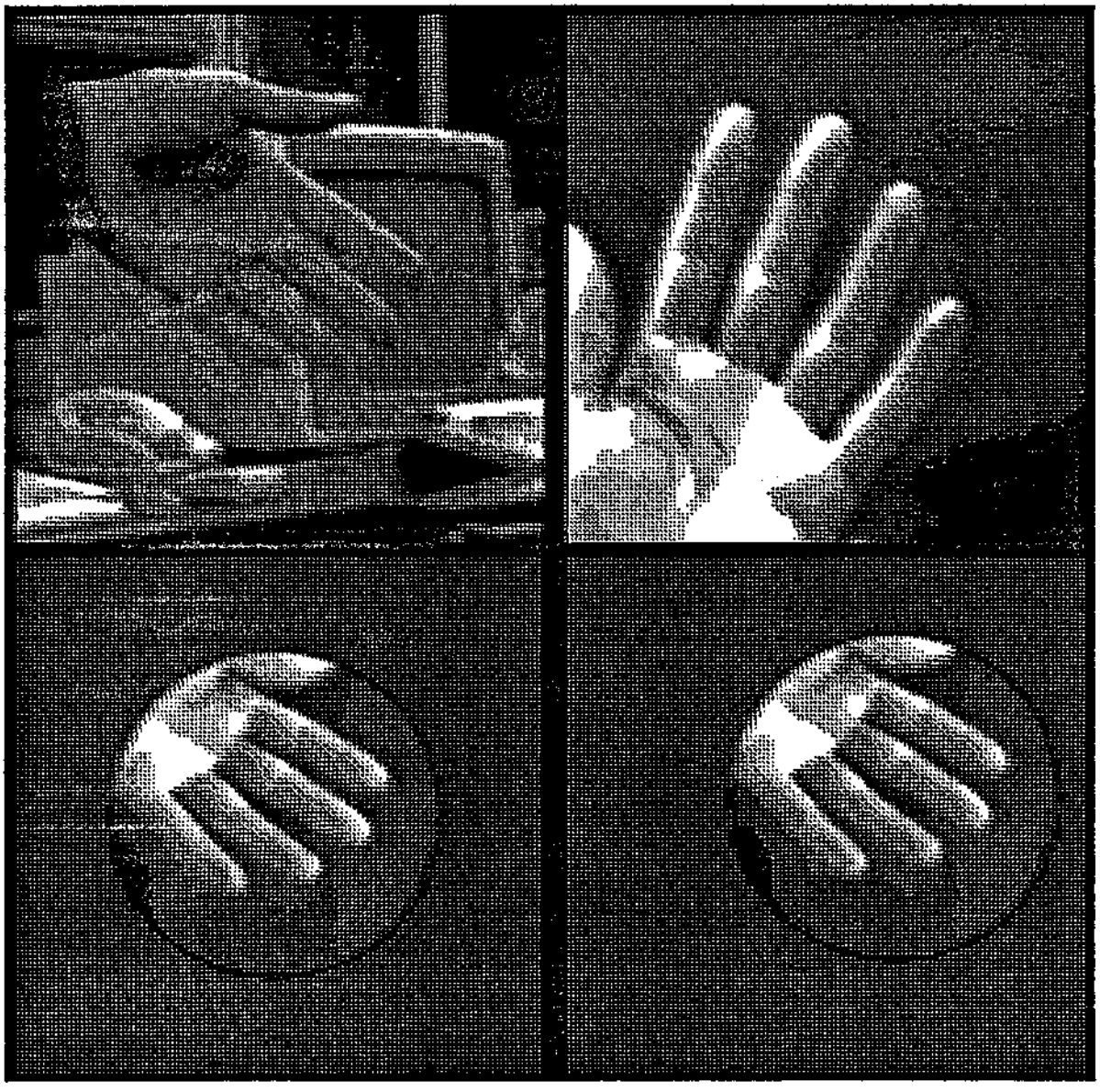

FIGURE 4: Matching two images in scale, orientation and position. Top row: original images. Bottom row: second original image transformed to match first original image; left: matched in scale and orientation; right: matched also in position. For more details, see appendix.

The LSI and biological vision

I have made no attempt to model biological vision, though a lesson has been taken from biology. Natural vision systems do not seem to use uniform sampling very much, and that certainly reinforces the idea that alternatives for computer vision are worthy of consideration. The human retinal receptor layout does not correspond to the LSI, but this does not preclude subsequent stages following a logarithmic pattern, and I would be delighted to be told of any relevant evidence. There are, for instance, intriguing indications in papers by Cavanagh (1985) and van Doorn and Koenderink (1982) that closely related representations may be used at some points in biological vision.

Finally, though, I must return to the point that what I have done is to open up consideration of new possibilities for computer vision. The principles on which image representation should be based remain, to me at least, obscure. If anyone can tell me how to determine the relative merits of the CSI, the LSI, spatial frequency representations or any of the many other possibilities, without just trying them out in a variety of tasks, I would be more than pleased to hear from them. So far, this aspect of computer vision has largely ignored biology, and as so often, it may be the case that we need to understand natural systems better if we are to get the right methods for artificial vision.

Acknowledgment

I thank Rudi Lutz for pointing out the paper by Funt.

References

Cavanagh, P. (1985). Local log polar frequency analysis in the striate cortex as a basis for size and orientation invariance. In D. Rose and V.G. Dobson (Eds.), *Models of the Visual Cortex* (pp 85-95). Chichester: John Wiley.

van Doorn, A.J. and Koenderink, J.J. (1982). Visibility of movement gradients. *Biological Cybernetics, 44*, 167-175.

Funt, B.V. (1980). Problem-solving with diagrammatic representations. *Artificial Intelligence, 13*, 201-230. Reprinted in R.J. Brachman and H.J Levesque (Eds.), *Readings in Knowledge Representation* (pp 441-456). Los Altos: Morgan Kaufmann.

Lee, D.N. and Young D.S. (1985). Visual timing of interceptive action. In D.J. Ingle, M. Jeannerod and D.N. Lee (Eds.), *Brain Mechanisms and Spatial Vision* (pp 1-30). Dordrecht: Martinus Nijhoff.

Østerberg, G. (1935). Topography of the layer of rods and cones in the human retina. *Acta Ophthalmologica, 13 Suppl. 6*, 1-102.

Appendix: Generating the figures

There is no hardware to obtain LSIs direct from images, so in my experiment to try out the line-finding method, I started with a CSI 512 pixels square. I resampled this to produce an LSI with 128 rings and 128 wedges, positioning the centre of the pattern arbitrarily at the centre of the image. In the centre of the LSI, the samples were much denser than in the CSI, so detail that should have been present was missing. In the periphery of the LSI the samples were much less dense than in the CSI. The mask I used was essentially a straight line, though with a few special effects thrown in: it was weighted towards the centre of the LSI, smoothed slightly, and differentiated with respect to ring number in order to pick out changes in grey level. None of these affects the basic principle, but they made the process more robust. I carried out the convolution using the fast Fourier transform, which is a computational trick to save processing time, and in fact I calculated the Fourier transform of my mask analytically. I ranked the peaks of the convolution result according to their absolute value, and plotted the lines corresponding to the few strongest on a picture of the original image. Figure 3 shows a typical result, for an image deliberately chosen to have a fair number of straight lines present.

Processing takes a modest number of seconds on a Sun 3 using POP-11, though doing the fast Fourier transforms and the logarithmic resampling with external Fortran routines. There are fast parallel implementations of all the algorithms used, which could be used if suitable hardware were available.

In my second experiment I obtained two CSIs, each 128 pixels square, of images on which fairly general matching could reasonably be expected to work. I enhanced the edges by convolution with a difference of Gaussians operator, and avoided the influence of the frames of the images by multiplying them by Gaussian-weighted windows. I then formed their autocorrelations using the fast Fourier transform, and resampled these using the LSI centred on the origins. Then the two autocorrelation LSIs were convolved together, again using the fast Fourier transform, with a few extras (smoothing and differentiation) thrown in for good measure. The global peak of the convolution result was taken to give the relative scale and orientation of the two images. Using these parameters, one of the CSIs was scaled and rotated (in fact by means of an intermediate LSI) to match the other, and a final convolution allowed the translations to be matched. It takes round about a minute to do all the processing on a Sun 3, but again it is easy to envisage very fast implementations.

Full details of these processes, and the mathematics of the straight line detection, will be given in future Cognitive Science Research Papers.