AI-1 VISION NOTES 1987/88

J A M Howe

D.A.I. TEACHING PAPER NO. 3

This supersedes Occasional Paper No. 48

Revised Edition 1988

DEPARTMENT OF ARTIFICIAL INTELLIGENCE
UNIVERSITY OF EDINBURGH

CONTENTS

INTRODUCTION

"How is human vision possible?"

Although the question has been addressed by philosophers and scientists for more than two thousand years, we cannot answer it with any degree of confidence. While the vast body of research work carried out by psychologists and physiologists has provided an enormous amount of information about both the human vision system's performance (by means of psychophysical experiments) and some of its neurophysiological structures, we still lack adequate explanations of how it functions. Some attempts have been made to provide functional explanations in the form of analogies with the technological devices of the day. An early example is the analogy between eye and camera; a more recent one would be an extension of the analogy from the still camera to the TV camera, and so on. In fact such analogies have been extremely influential: the analogy between eye and camera has dominated twentieth century psychological research into vision. Unfortunately, it has also misled it due to a confusion between structure and function. From a structural point of view, we can readily map between parts of a camera and parts of the human eye - both form 2-D images by means of lenses which bring light to a focus on a surface (film or retina). In the case of the camera, we all understand the next step - we take the film out of the camera, develop it, make prints of the frames and look at them. But these prints are only meaningful when an intelligent agent looks at them and makes an interpretation. In the case of the eye-camera analogy, what is the equivalent of processing the film and interpreting the prints? A much favoured explanation was the notion of an "inner screen", situated in the visual brain, on to which a "picture" of a scene was mapped. While this dealt with part of the problem, the print, it did not address the other part, the interpretation of the print. If we pursue the analogy, we can see that the logical conclusion is that there is a second observer inside the visual brain who looks at the 'picture on the inner screen'. Presumably, the outcome of his perception is another picture on his inner screen, which is viewed by a third observer who is inside the visual brain of the second observer who is inside the visual brain of the first, and so on. In other words, the attempt to explain the process of vision by means of an analogy between eye and camera leads to an infinite regress.

While this logical fallacy ought to have disposed of the eye-camera analogy for once and for all, instead its influence has pervaded 20th century visual psychology. Since the eye has a two-dimensional light sensitive surface, psychologists have regarded space perception, i.e. 3-D perception, as paradoxical. This has led to the question "what additional information and what properties of the two-dimensional image give rise to three-dimensional experiences?". Until recently, the experimental psychology of space perception has been dominated by this problem, the quest for additional pieces of information which, when added to the flat image at the back of the eye, make 3-D perception possible. What this information is need not concern us. It is sufficient to note that the concept of perceiving 3-D space through the use of supplementary information is a by-product of the eye-camera analogy. The job of combining the supplementary information with the two dimensional images received from the eyes to promote judgments about the relationships of objects in depth is akin

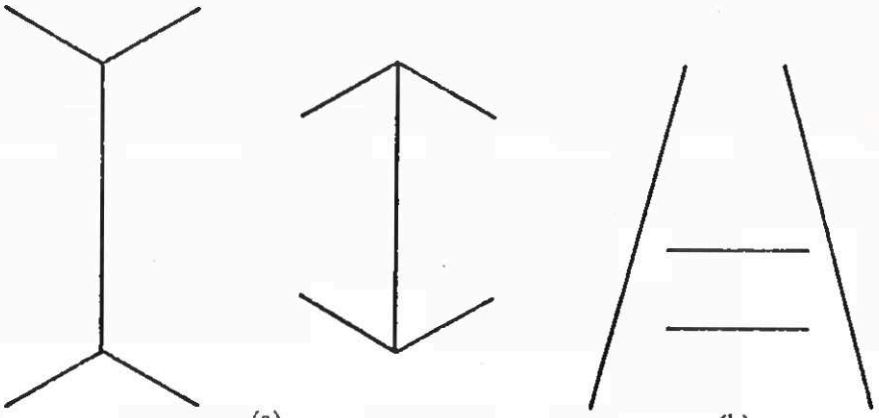
to that of an air-traffic controller. The controller observes a radar screen that provides a two-dimensional representation of the air space under surveillance. He must obtain information about the third dimension, altitude, from the signal sources such as radio transmissions from the aircraft. He has to combine the 2-D positional information from the radar set with the altitude information to make judgments about the paths that should be taken by the aircraft.

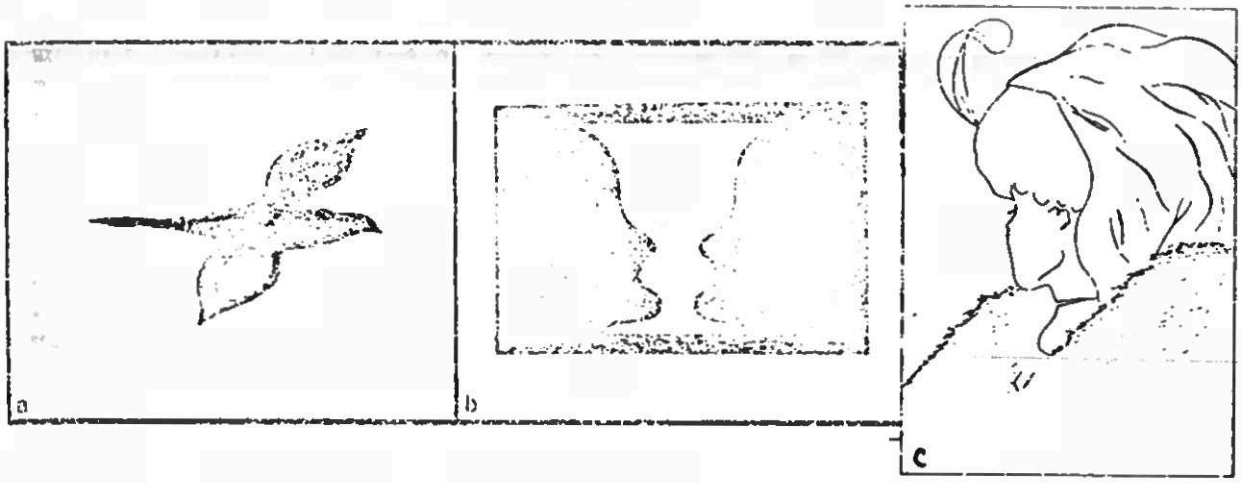
To interpret, or not to interpret.

It would seem that an adequate theory of vision has to explain the interpretative processes that give meaning to the information extracted from the physical world by the eye.

However, not all vision psychhologists would agree with this statement. In particular, J.J. Gibson favoured what is often described as a theory of 'direct' perception. Starting from the question, *How does one obtain constant perceptions in everyday life on the basis of continually changing sensations?*, Gibson argued that vision was concerned with the recovery of valid properties of the physical world (called "invariants") from the ever changing sensory information, whether due to changes in the intensity of stimulation or to movement of the observer. Thus, he wrote that the "function of the brain, when looped with its perceptual organs, is to decode signals, not to interpret messages, nor to accept images, nor to organize the sensory input or to process the data, in modern technology. It is to seek and extract information about the environment from the flowing array of ambient energy." As Gregory has pointed out, (Gregory, 1981), Gibson's explanation is reminiscent of one of the earlier theories, advanced prior to the discovery that light formed images of objects on the retina, which suggested that objects gave out 'husks', or 'simulacra'. which acted as intermediates when we see (the same idea can be found in the 'sense-data', proposed by the philosophers Broad and Price).

The contrasting position, held by Descartes, Helmoltz, Gregory and others is that perception is a constructive process, involving inferential reasoning over sensory information and knowledge of the world stored in memory. This explanation was motivated by many observations that perceptions frequently do not correspond to their physical correlates. Obvious examples include brightness constancy (the surface of an object appears evenly lit, even though the physical illumination varies across it), size constancy (the perceived size of an object is not determined by the size of its retinal image, and shape constancy (similar to size constancy). Less obvious examples are the many and varied 2-D and 3-D optical illusions, ambiguous shapes, and so on, made so popular by Gregory, and used to underpin his argument that the human visual system's task is to evaluate alternative visual hypotheses against the available sensory information. Some of Gregory's favourite examples are shown below. The upper illustration includes (a) the Muller-Lyer and (b) Ponzo illusions; the lower includes (a) Hawk/Goose, (b) Vase/Face and (c) Wife/Mother-in-Law.





One of the most dramatic examples of the interpretative capability of the human visual system is Ames Window. He designed a trapezoidal window that resembles a rectangular window viewed from an angle. When viewed from a sufficient distance with one eye only, the rotating window appears to be oscillating. The direction of rotation is correctly reported during the 180^0 rotation for which the longer side of the window is close to the observer, and incorrectly reported during the 180^0 for which the shorter side is closer. When a rectangular window is rotated with the same viewing conditions, the direction of rotation is correctly reported throughout the 360^0 cycle. Why does the motion reverse in the former case, but not in the latter? Obviously, the shape transformation produced by rotating the trapezoid differs from the shape transformation produced

when a rectangular window is rotated (or more likely when we walk past rectangular windows). In theory, the visual system would explain this difference in two ways, (1) by accepting continuous rotation and allowing the structure to deform, or (2) by accepting a rigid structure and varying the rotation. In practice, as we shall see in due course, the visual system prefers interpretations which favour rigidity, so the latter interpretation is preferred.

If we go one step further, and place a solid rod through the mullions of the window, a further paradox is created. While the window oscillates, the rod turns in one direction only. This is impossible. In general, the mind refuses to accept that the rod can pass through the solid mullions of the window. Instead, most people perceive the rod twisting and bending around the window structure, even though they are well aware that it is a solid object. In this case the brain is prepared to abandon the rigidity assumption as applied to the rod in favour of maintaining the rigidity of the larger, more complex object, the window.

These two types of theory are at the opposite ends of the dimension and, in some sense, represent extreme positions. It seems likely that Gibson adopted his stance through a desire not to fall into the "inner eye" trap. While it cannot explain many of the phenomena cited by the proponents of the knowledge-based approach, Gibson's approach did draw attention to many, previously ignored features of optical images. Those favouring a knowledge-based approach, on the other hand, have not been able to explain how knowledge is stored. how it is invoked, how it is reasoned over, whether the gathering of information is affected by the internal processing, and a host of other equally apposite questions.

The role of computational vision.

Workers in the area of computational vision in AI are also trying to build a theory that explains the phenomenon of human seeing. However, the type of theory that they favour is called a process theory. What this means is that the theory should propose effective procedures for interpreting the visual data captured by the eye, with the objective of generating some specified perceptual output.

Some functional requirements of such procedures are as follows:

* Geometric modelling. Determine the 3-D configuration of surfaces and objects in a scene, including the viewer's location

* Photometric modelling. Determine the location and nature of the illumination sources and the corresponding shadowing and reflectance effects induced in an image by the scene.

* Scene segmentation. Partition the scene into coherent sub-units that can be independently analysed and identified.

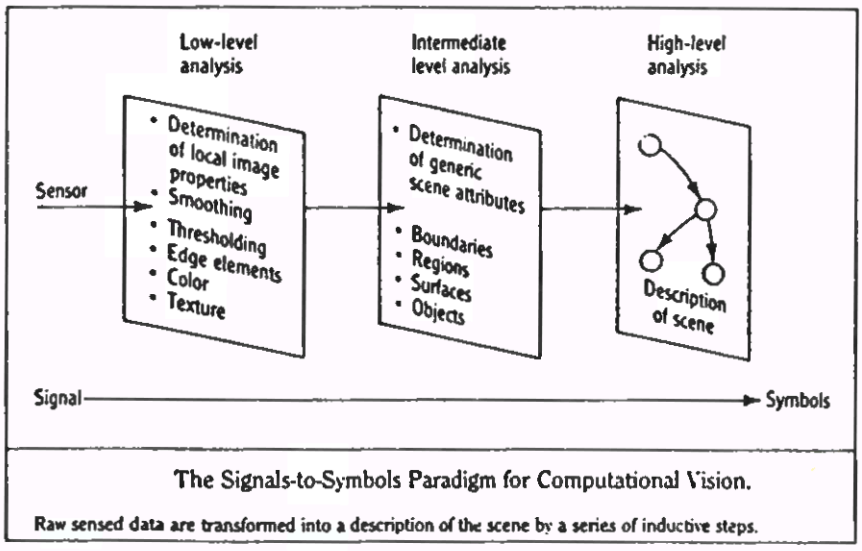
* Naming and labelling. Identify objects visible in a scene either as members of known object classes, or as known individuals. Determine physical attributes of recognised objects.

* Relational descriptions and reasoning. Determine the relationships between objects in a scene. Determine how they can be re-arranged to achieve some specific purpose.

* Semantic interpretation. Determine the function, purpose, intent, etc. of objects in a scene.

The digital computer is the tool favoured by AI research workers for building a process based theory. To identify its role, we can make an analogy between building and testing symbolic models on a computer, and building and testing physical models using a mechanical construction kit, such as Meccano., Building a mechanical mechanism is done by selecting appropriate mechanical parts from the kit, e.g. using an electric motor to drive a mechanical model of a car. Building a symbolic mechanism is also done by selecting appropriate symbolic parts, i.e. commands, from the list of symbolic commands provided by the programming language, putting them together to form a program, the structure, and testing that program by running it in the computer. In both cases, the mechanism, physical or symbolic, will generate an action sequence, or behaviour, which is open to inspection and interpretation by the designer. The extent to which the mechanism's behaviour satisfies the programmer's expectations in some sense tests the adequacy of the underlying design of the mechanism. If this is close, modest changes to the mechanism may be sufficient to achieve the expected performance; but if it is wide, a reconceptualization might be required. This, then, is the methodology of computational vision.,

Of course, for a digital computer to deal with the visual world in this way, the signals acquired by its imaging device must be converted into symbols. The signals-to-symbols paradigm is illustrated below (from Fischler & Firschein, 1987) where a series of inductive steps employing progressively more abstract representations transform raw sensory information into a meaningful and explicit description. These steps are partitioned into three broad categories, depending on the kind of modelling required for analysis purposes: low-level analysis is based on local image properties, intermediate-level analysis uses global properties, and high-level analysis employs semantic models and relationships.

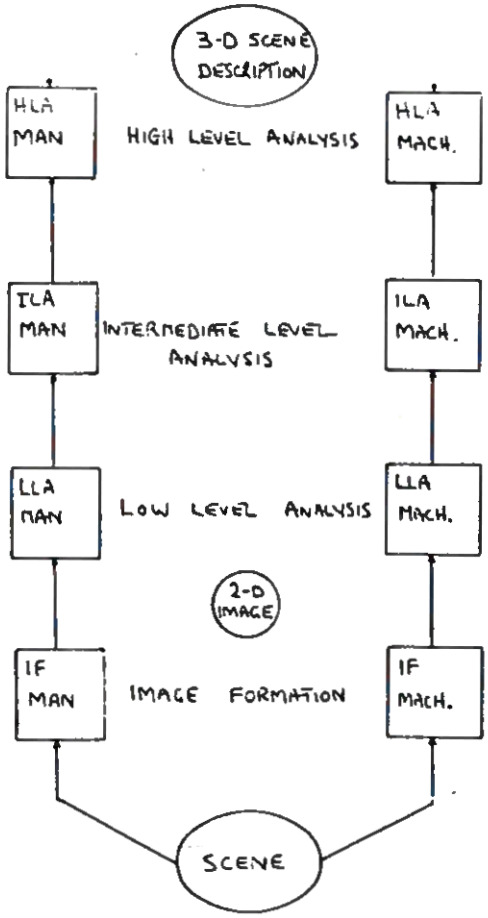


The Signals-to-Symbols Paradigm for Computational Vision.

Raw sensed data are transformed into a description of the scene by a series of inductive steps.

Objective of the course.

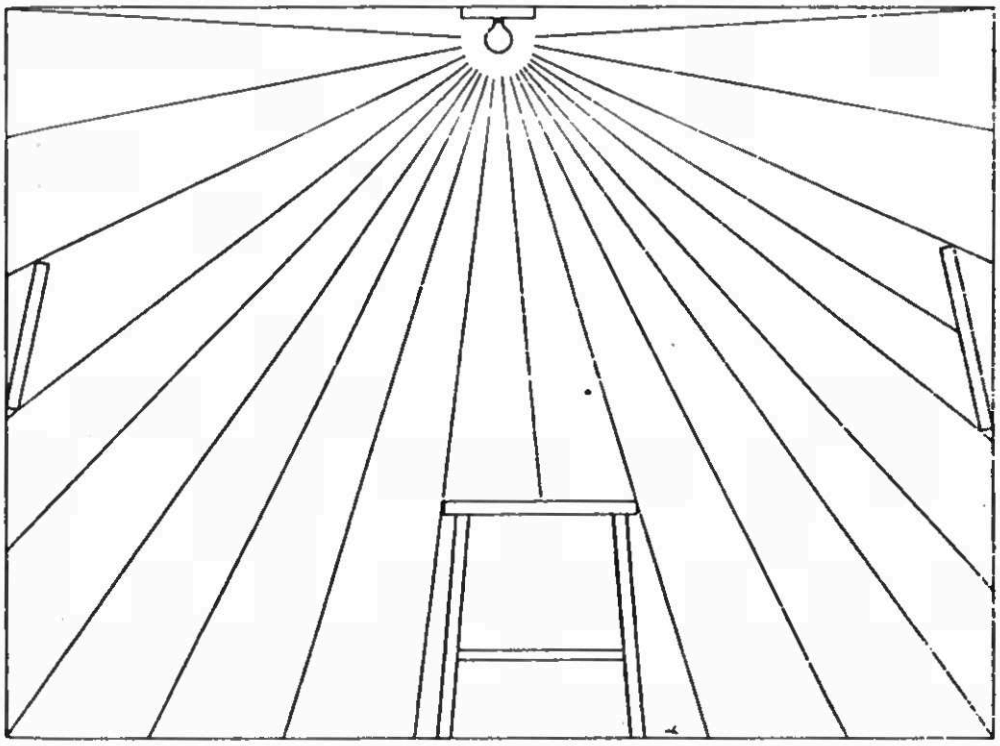
The purpose of this course is to use the computational approach to explain how we come to see a three-dimensional world containing objects that have stable sizes and shapes. To assist us, we will make a simple comparison between man and machine, as shown below. The ultimate goal of computational vision is to achieve a symbolic interpretation similar to that of man for the same input pattern. The right hand side deals with the artificial vision, whereas the left hand side deals with biological vision (in man). Since relatively little is known about artificial systems, the kinds of processes employed by the latter system will be used to throw light upon possible biological analogues.



The material is broken into three sections, corresponding to the three levels of analysis described above. The first level will be concerned with computational procedures that illuminate the neurophysiology of vision; the second level will examine some facets of intermediate level analysis from a computational standpoint, and the third section will deal with computational aspects of visual recognition.

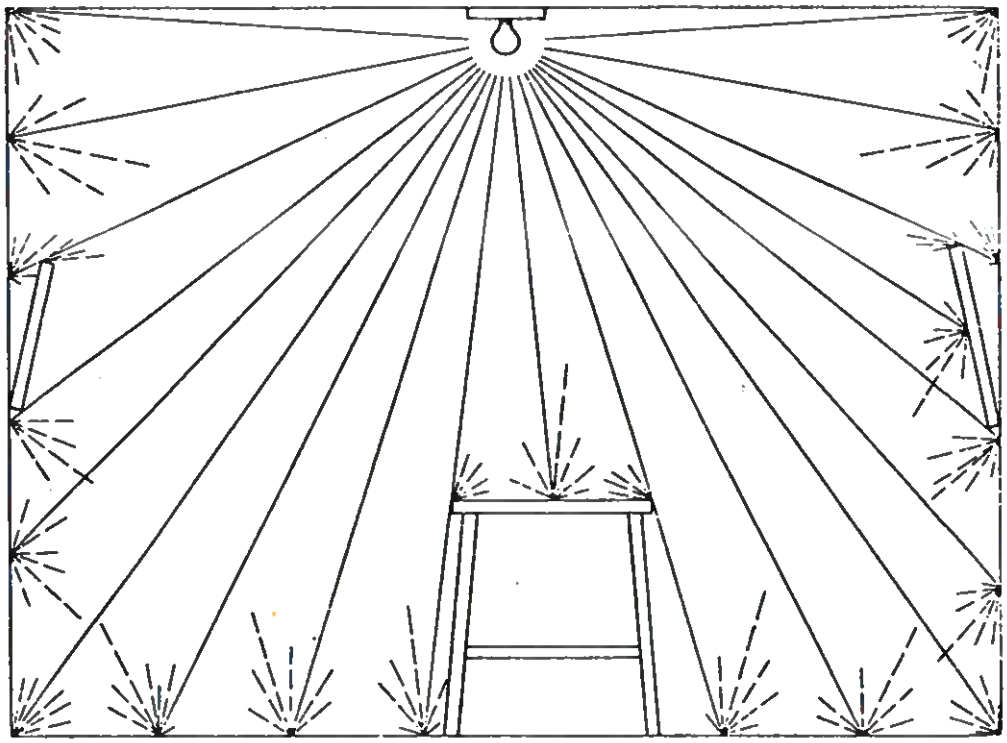
LIGHT AS INFORMATION

We will begin by considering how light transmits information about the structure of the environment to the eye of the perceiver. To help us, we will distinguish between radiant light, that is light emitted from an energy source such as a sun or star, and ambient light, that is, light reflected by the surfaces in the environment. Ambient light is much more complex than radiant light, so we will start by briefly considering radiant light. Radiation is shown below:



The light rays diverge from the source, the lamp, and if they were in an empty space they would continue indefinitely. Only a very few rays are shown, but there are infinitely many rays present. Unless reflected, all an observer would see is a luminous spot at the source.

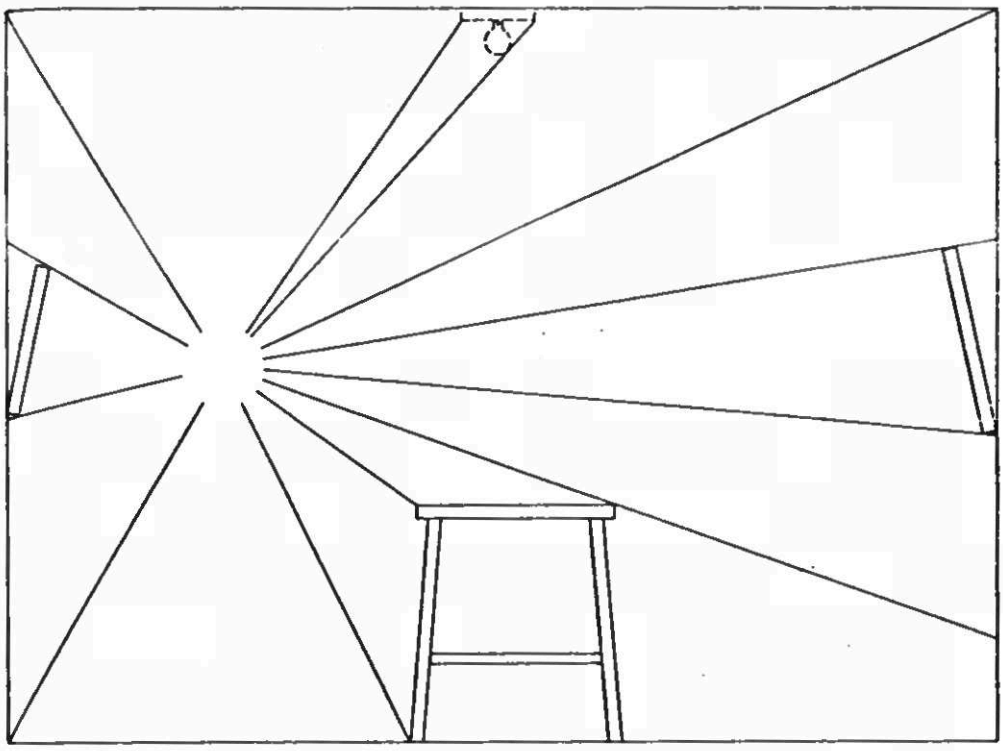
Radiant light contains information in the form of a distribution of wavelengths. But while optical instruments can determine whether a particular mix of frequencies has been produced by an incandescent source, or a fluorescent source, the human eye cannot interpret the distribution of wavelengths, nor can it measure their absolute intensities. This is not the kind of information that an eye can pick up. Instead, the eye is designed to make sense of ambient (reflected) light and differences in light intensities. Reflection is shown below:



If the surfaces are not smooth, the rays of light are scattered in various directions, depending on the micro structure of the surface at each position. For example, the rays reflected by a matte surface, i.e. a rough surface, are more scattered than the rays reflected by a polished surface, i.e. a smooth surface. Indeed, as a surface gains in gloss, shine or lustre so the amount of scattering is reduced. The limiting case is a mirror, when scattering is eliminated.

When there are a number of surfaces facing each other, the light bounces from surface to surface endlessly. At this stage the environment is said to be illuminated. An infinitely dense network of light rays is created - in other words there are intersecting rays at every point in the space enclosed by the surfaces. The converging and diverging rays cannot be represented in a diagram, but must be imagined.

Now we will consider a point in a room with facing surfaces, as shown below. The radiant light is omitted:



Instead, the lines in the diagram correspond to the edges and corners of surfaces facing in different directions. They are the boundaries between bundles of light rays. The reason for these boundaries is that the surfaces of the room reflect different amounts and different colours of light to a convergence point.

For example, two adjacent surfaces which have the same microstructure but which are set at different angles of inclination to the light source will project different intensities. Two adjacent surfaces which have different microstructures or different pigmentation will project different intensities, even if they are set at the same angle of inclination. In both cases there will be a variation in luminance where the two surfaces meet. This variation is seen as an edge.

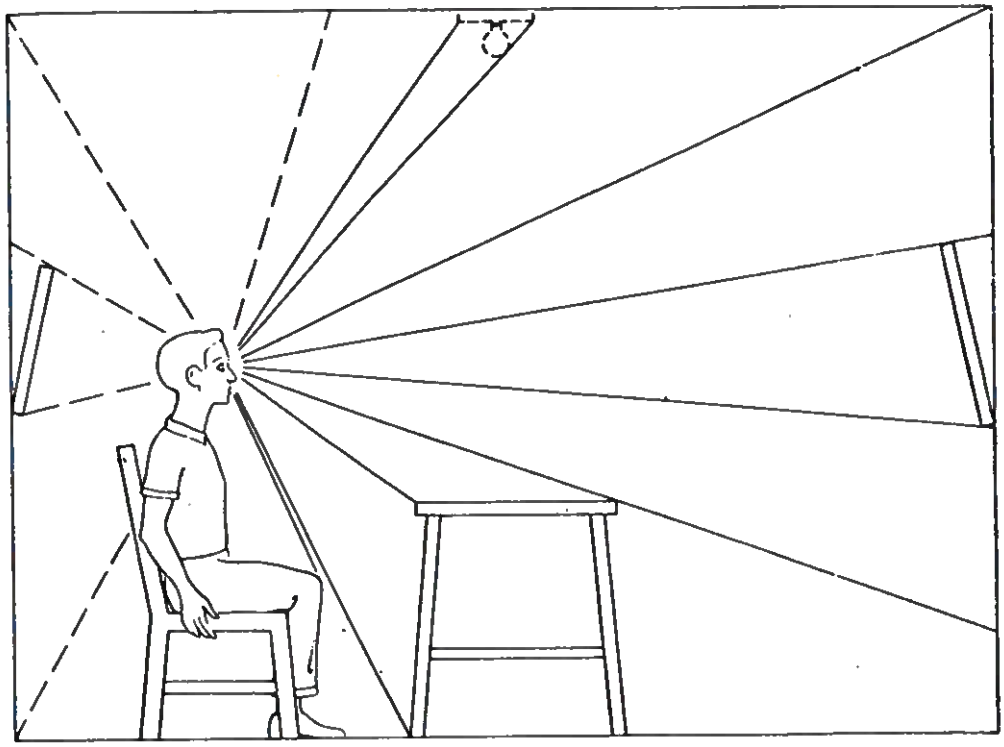
So, in summary, ambient light is not a random collection of light intensities, but is an organized collection of intensities, the organization being imposed by

(1) the physical inclinations of the surfaces

(2) the reflectances of the surfaces, and

(3) the spectral characteristics of the surfaces.

Now we introduce an observer, as shown below:



His eye admits a sample of the total ambient light. Now, for every edge in this sample there is a corresponding edge on the retina of the eye. In the case of a stationary observer, these edges are represented by variations in the luminance at different positions across the retina.

If information about structural dispositions is conveyed by luminance patterns, seeing that structure should boil down to detecting the presence of these luminance variations in the sample of ambient light at the retina. For example, an abrupt change in luminance might be interpreted as signalling the edge between the surface of an object and its background. A gradual change in luminance, on the other hand, might represent a convex or concave edge between two surfaces of a given object.

But unfortunately in the real world the situation is not so simple. Luminance variations do not necessarily represent edges at all. They can also represent highlights, shadows, illuminations, gradients across surfaces, dirty surfaces, scratches, and so on. In other words, the information in the luminance distribution is highly ambiguous. Yet we rarely confuse a shadow edge with a real edge, so the human visual system is able to cope with this ambiguity. This is a crucial problem which we have to solve in due course.

I. LOW LEVEL ANALYSIS

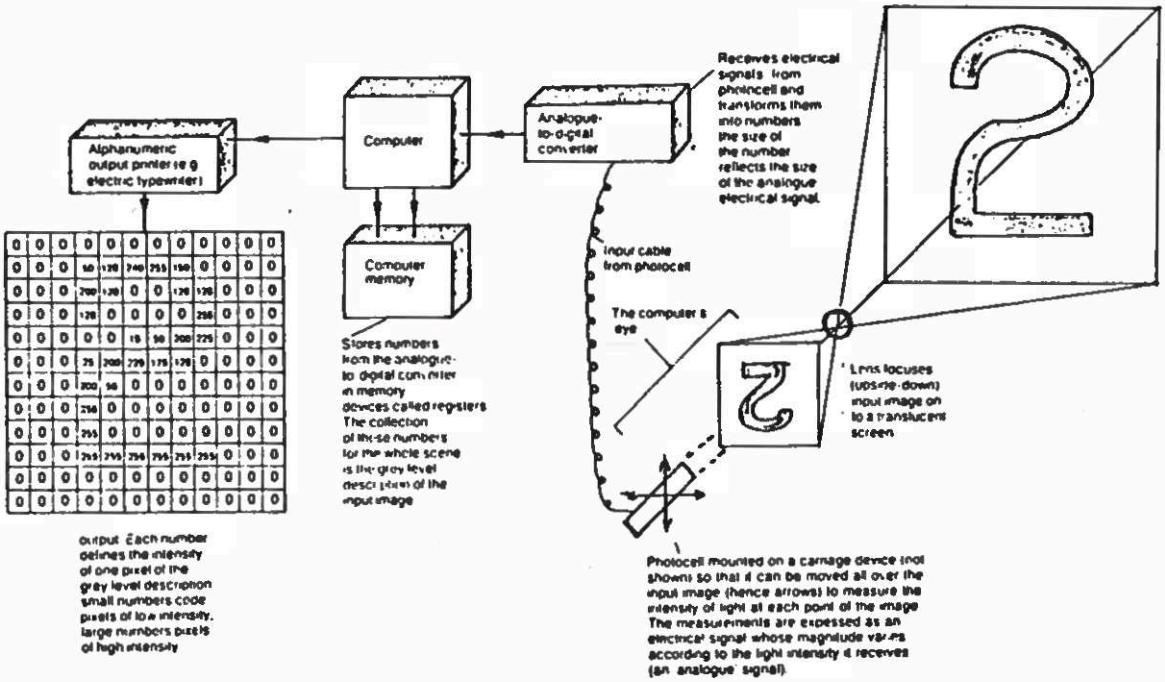
ARTIFICIAL EYE

Quantization

It is time we turned our attention to artificial eyes. We will consider how an artificial system might build a symbolic representation of an object in a scene, where the symbols denote candidate edge prints. After that, we will try to use what we have learned from building the artificial system to help us understand what kinds of local computations might be made at the level of the human eye.

For the time being we will be considering a single block on a table top. The light reflected from the block is focused on to the sensitive surface of a phototransducer, a device for converting light into an electrical signal, where it forms local patches of light and dark. These variations in luminance carry the information about the features of the block in the scene. Recollect that any edge or contour is specified by this variation. For example, an abrupt change in luminance might indicate a boundary edge between object and background, whereas a gradual change in luminance might indicate a convex or concave edge between two of its surfaces.

The first step is to convert scene luminance information into an array of numbers in the computer's memory. This is done by moving a small window over the image and measuring the average luminance level within the window by means of a light sensitive device called a phototransducer. The process of moving the window over the image is called scanning. A simple image acquisition system, using a mechanical scanning device, is shown below:



The phototransducer is mounted on a moving carriage which travels across the image from left-to-right, and back again. Each time it changes direction, it also moves down the image by a small amount. The output from the phototransducer is a continuous electrical voltage signal whose size varies according to the luminance level at its input. This continuous signal has to be converted into an array of numbers by sampling the signal at regular time intervals (corresponding to regular spatial intervals over the image) and then approximating the measured voltage values at the sampling points by the nearest numerical values in a pre-defined range of integers. The sampling points, the 2-D positions in the image at which the samples are taken, are known as picture elements, commonly abbreviated as pixels. The process of converting the signal into numbers is often referred to as quantization.

The output from quantization is an array of numbers, usually referred to as the grey-level description of the image.

The number of image points, or pixels, making up a computer's grey level description, varies according to the capabilities of the computer (for example, the size of it memory) or the needs of the user. For example, the use of a dense array of pixels will require a large memory store and produce a grey level description that picks up very fine detail. For example, an ordinary domestic TV set produces an image in pixel form with an array size of 625 x 625 pixels. The individual pixels are so tiny that they cannot be readily distinguished (unless a large TV screen is viewed close to). On the other hand, it may be necessary to use large pixels, each of which represents a large local area of the input image, in which case a full-tone printout produced to the same scale takes on a block-like appearance.

The word precision is used to refer to a grey-level description's ability to represent fine detail in an image. But even if the processing is precise enough to capture fine detail, some of the grey-level information might be lost or altered in the process of placing the digitised image in the computer. This is the accuracy dimension of the problem. Such losses may be due to imperfections in the performance of the transducer, e.g., non-linear scan; to errors introduced by the process of converting the image from analog to digital form, or simply to electrical noise in the analog circuitry. This noise is intrinsic noise, i.e., it is due to the operation of the mechanism, it is not due to external factors. The effects of intrinsic noise can never by fully eliminated. Notice that the most common form of error introduced by intrinsic noise consists of small isolated regions of the picture that are much brighter or darker than they should be. To eliminate many of these problems, a smoothing operation can be used. Basically, smoothing operations rest on the assumption that the actual scene consists of areas that are very much larger than the areas represented by a single point. Accordingly, picture points that differ markedly from their immediate neighbours are errors that ought to be removed.

The following procedure describes a simple smoothing operator:

> If any point in the picture is brighter than all of its eight immediate neighbours, its luminance value is reduced to make it the same as the brightest of its neighbours; if any point in the picture is dimmer than any of its eight immediate neighbours, its luminance value is increased to make it the same as the dimmest of its neighbours.

Notice that this operator is conservative in the sense that it removes some of the noise without reducing the amount of information in the representation. In particular, it eliminates isolated noise points, but has no effect upon noise that occupies two or more adjacent pixels.

A simpler, more liberal smoothing operator that would reduce the significance of larger regions of noise is:

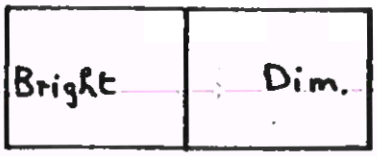
> Replace the luminance value of each point by the average of the luminance values of its eight immediate neighbours.

Unfortunately, the application of this operation to every point in a picture will have the effect that every edge will be blurred. Indeed, several successive applications would wash out the entire picture. Clearly, therefore, smoothing operators are useful, but must be carefully chosen to try to eliminate whatever kind of intrinsic noise is present in a digitised image, without also removing significant features.

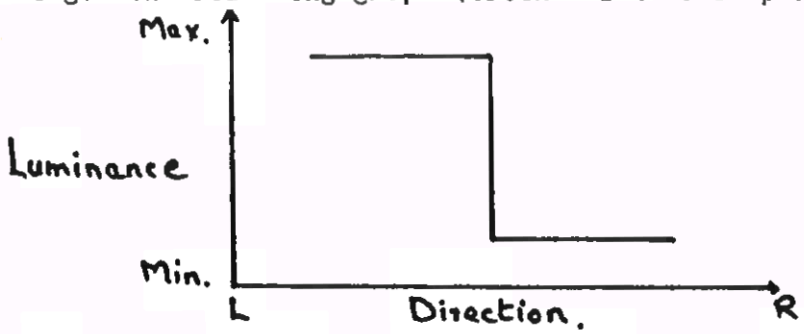
Selecting edge points

We turn now to consider the next step: identifying pixels which are hinting at the presence of edges in the grey-level description i.e. candidate edge points.

Suppose we have two adjacent regions, one bright and one dim:

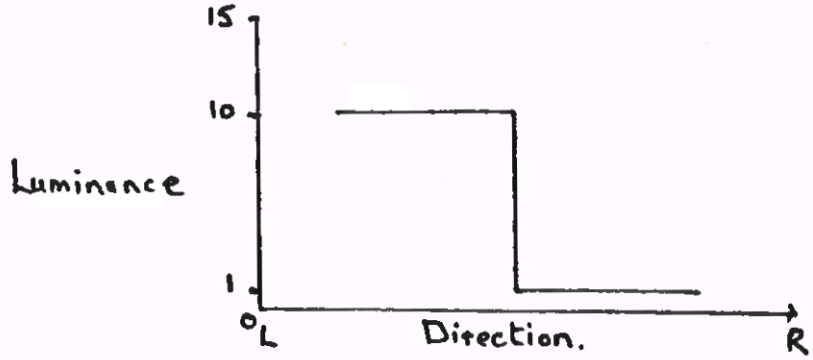


If we make a one-dimensional plot of the luminance values from left-to-right, we get the following graph (often called a step function):

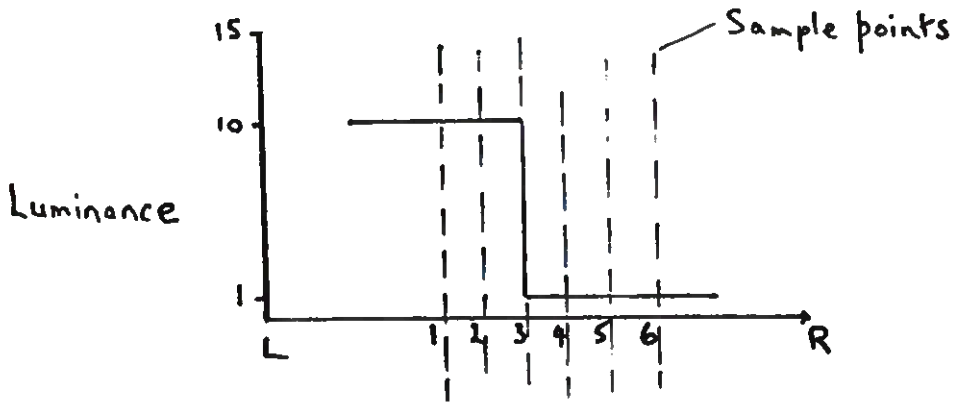


We are interested in detecting the discontinuity between the light and the dark regions. The effect of the differences in absolute level of luminance can be eliminated by taking the first derivative of the step function.

Suppose the maximum luminance value of the function is 10 and the minimum value is 1. The one dimensional graph becomes:

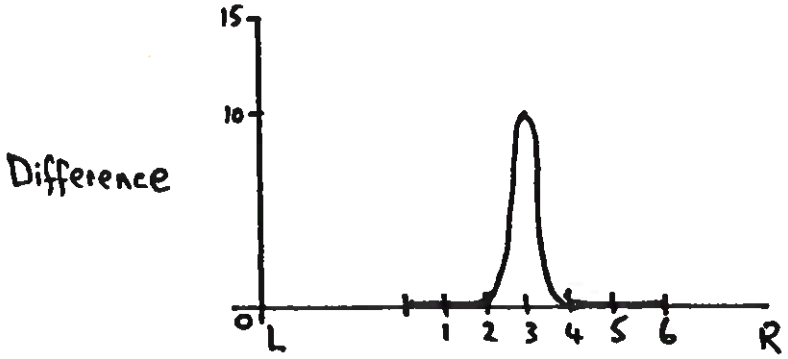


Now, suppose we sample the function in one dimension, as follows:

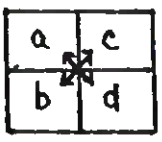


The first derivative is calculated by taking differences. Starting with samples 1 and 2, the value of the function is 10 in both cases, so the difference is 0. It is also 0 for samples 2 and 3. But when we get to samples 3 and 4, the difference is 9. Moving to samples 4 and 5, we get a difference of 0, and again when we take 5 and 6.

We can plot the first derivative in one-dimension as follows:



In an image, the values are changing in <u>two-dimensions</u>, not one-dimension, so we want to take the differences in orthogonal directions. A suitable operator for two-dimensional differencing is the following (usually referred to as the Robert's cross operator):



$$Gx = |a-d| + |b-c|$$

Suppose we have the following fragment from a grey-level description:

Column	A	B	C	D	E	F	G
Row 1	1	1	2	4	5	5	5
2	1	1	1	5	6	5	5
3	1	1	1	4	6	5	5
4	1	1	1	5	5	6	5

If we take the cells A1, A2, B1 and B2, and apply Robert's operator, we get

1	1
1	1

$$Gx = (1-1) + (1-1)$$
$$= 0$$

If, however, we take cells C1, C2, D1, D2, and apply the operator, we get

2	4
1	5

$$Gx = (2-5) + (1-4)$$
$$= 6$$

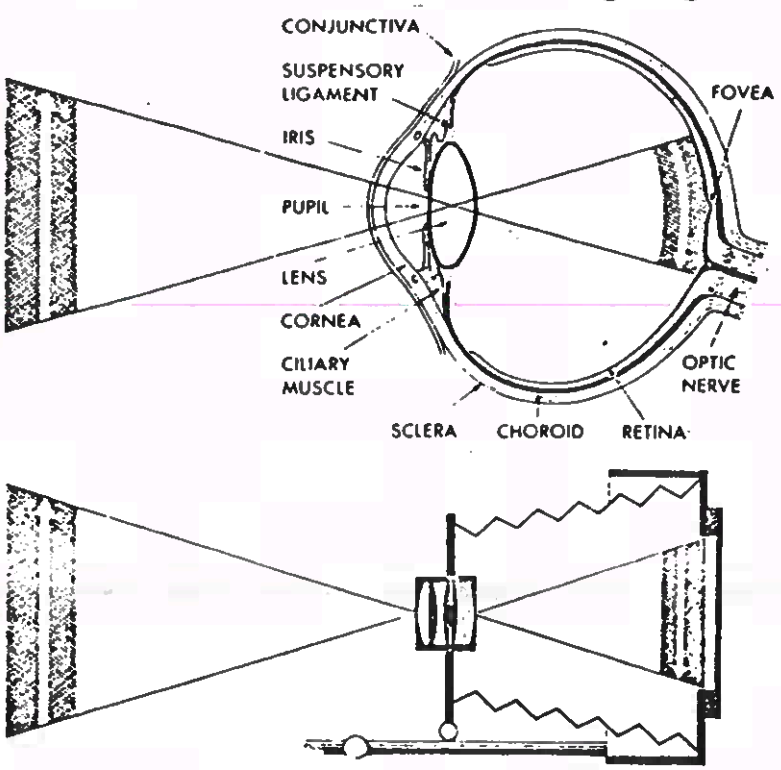
If we look at the data, we can see that there are two relatively homogeneous areas: one is to the left of column D, viz. columns A-C, and the other is to the right of column D, viz. D-G. The discontinuity lies between C and D. The presence of this discontinuity is suggested by the high value returned by the cross operator when applied to the cells C1, C2, D1 and D2. Similar high values would be returned for cells C2, C3, D2, D3; C3, C4, D3, D4, and so on.

In practice, diagonal differences are calculated between adjacent pixels across the whole image. The resulting values are stored in a new array (sometimes called the differential description). This new representation contains the candidate edge points. Obviously, low values correspond to areas of uniform luminance (e.g. background, surfaces of objects), whereas high values are most likely to be associated with changes in the luminance caused by edges in the original scene.

THE HUMAN EYE

Structure

In a typical environment, objects are diffusedly illuminated, that is, light rays are reflected off objects in all directions. How does the eye capture them? One answer is that the human eye captures light rays in similar fashion to a camera. Consider the following diagram:



Light enters the eye through the cornea, a tough protective membrane which acts like a convex lens, bending the light rays together. Behind the cornea is the iris, a coloured annular muscle which opens and closes like a camera diaphragm. The small round aperture in the middle of the iris is the pupil. To record a scene in full detail, a camera film must receive a particular dose of light energy. This is done by setting an exposure time, and varying the size of the aperture. The choice of exposure time and aperture size is a trade-off. A long exposure time, using a small aperture, will produce a better defined picture since the amount of scattered light is reduced by the small aperture - but the problem is keeping the camera steady to prevent blurring due to camera movement. Shortening the time minimises problems of camera movement, but the aperture must be opened up so the quality drops off due to increased light scatter within the camera. In similar fashion, a retinal cell must receive a minimum amount of energy before it will fire. So here, too, there is a trade off between intensity and time, but there is one substantial difference. Whereas the sensitivity of a film is fixed, the sensitivity of the retina varies in a manner that relates to the prevailing lighting conditions. This process is known as adaptation, and it is this process which enables the eye to detect variations in luminance over a wide range of light levels. In the case of a camera, one has to use films of differing sensitivity for photographs taken at widely different levels.

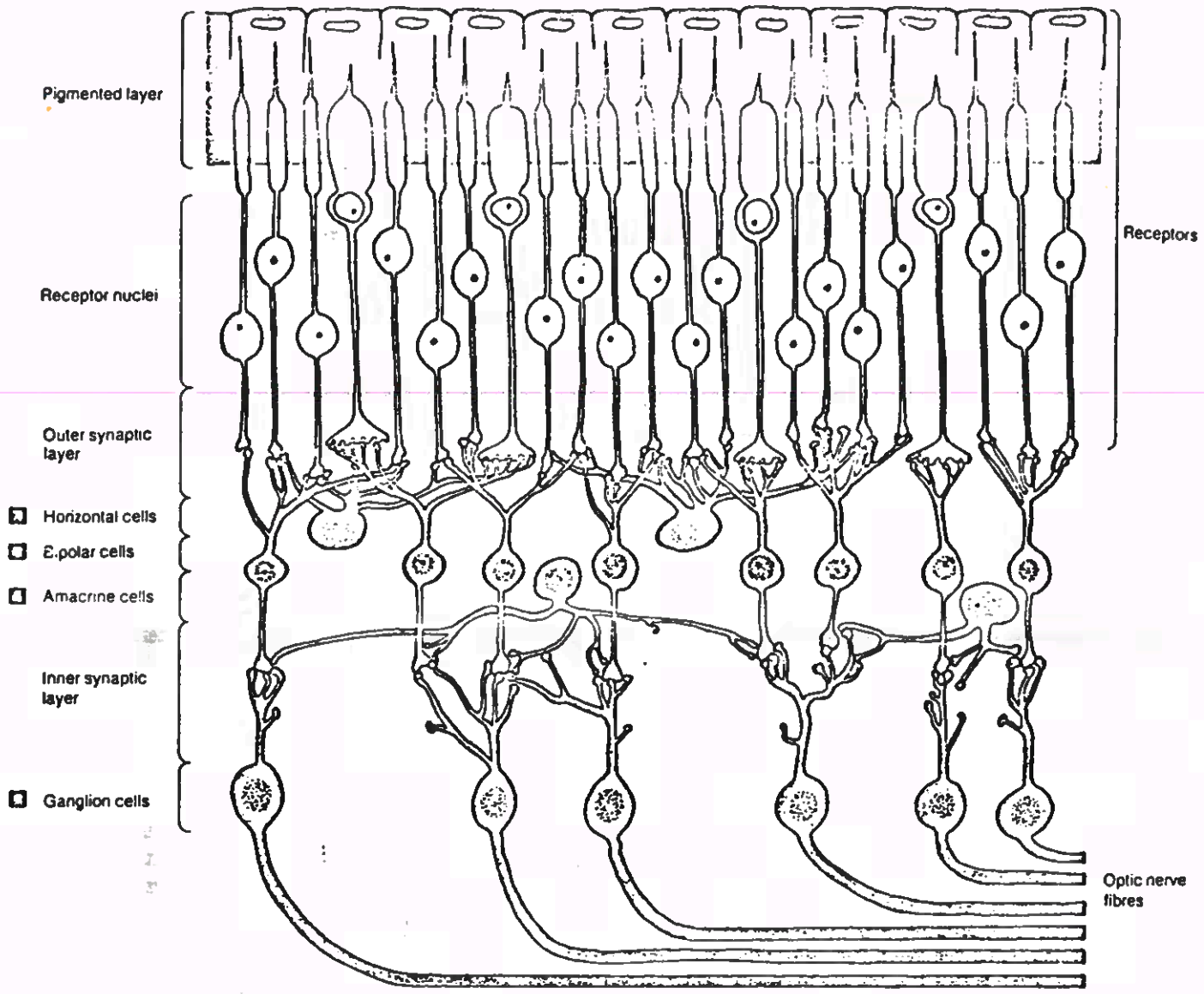
Going back to the pupil, we can see that its role is confined to making minor variations in the sharpness of the image: unlike the camera diaphragm, it is not an essential image forming component.

Light passes through the pupil to the lens. To focus light rays in a camera, the lens-to-film distance is altered slightly. In the eye, the lens focusses light by changing its thickness, and hence its refractive power, i.e. ability to bend the rays. This is achieved by the action of a muscle/ligament arrangement which alters the curvature of the front face of the lens.

But there is yet another difference between the camera lens and the lens in the eye. Camera lenses are multi-layer lenses, carefully shaped to overcome various problems like chromatic and spherical aberration. Chromatic aberration occurs if a lens is unable to bring light rays of differing frequencies to a focus at the same depth plane. Spherical aberration refers to shape distortions, caused by slight variation in the curvature of the lens. Unlike the camera lens, the lens in the eye suffers from both defects, thus the image quality at the retina is less than that at the film plane in a camera.

Since the eye is a living thing, it has to be supplied with nourishment. The spaces between cornea and lens, and lens and retina, are filled with a clear glutineous mass which provides the nourishment, besides helping the eye to keep its shape. The energy in these fluids is replenished by a network of blood vessels inside the eye, between the lens and the retina. With the exception of a small area in the middle of the eye which is not obstructed by blood vessels, light reaches the eye by passing through these blood vessels. That we do not see shadow images from them is due to the fact that light always reaches the retina from precisely the same direction, down the optical path, so that receptors in the shadow of obstructions can selectively adapt their sensitivity. This adaptation prevents the fixed obstructions being seen provided the shadow images remain stationary on the retina. In fact, the retinal blood vessel structures can be seen by dark-adapting the eye, by placing a small torch in close contact with the shut lid of the eye, near its side, and by moving the torch with regular oscillation. This projects a shadow image of the structure on to non-adapted receptors. So here is another way that the eye differs from a camera: there are no obstructions between the camera lens and the film during the picture taking process.

But blood vessels are not the only structures interposed between lens and the photo-receptive layer of the retina: the retina is actually "inside out" so light also has to pass through retinal structure before stimulating these cells. These are shown in the drawing, opposite.


Pigmented layer
Receptor nuclei
Outer synaptic layer
Horizontal cells
Bipolar cells
Amacrine cells
Inner synaptic layer
Ganglion cells
Receptors
Optic nerve fibres
Light

Let's look at this structure in a little more detail. First of all, the retina of the eye has 130 million receptor cells. These extend approximately 100° from the visual axis. These cells are distributed much more thickly at the centre of the retina than at the edge, falling from a density of 160,000 per square millimetre at the centre to 1,000 per square millimetre at the retina's edge. Again, the eye differs from a camera where the film is equally sensitive all over: this variation in sensitivity across the retina accounts for our ability to perceive sharp detail in front of our eyes but only crude shapes at the side.

The structure is even more complex since there are two kinds of receptor cells, namely cones and rods. Although there are only about 6 million cones, compared with approximately 125 million rods, our normal day-time colour vision is thought to be due to the action of the cones which are most densely packed in a 1° sized central region of the retina, called the fovea centralis which is rod free.

The receptor cells convert light energy into electrical signals in nerve fibres. We will look at the properties of cells in due course. For the moment, notice that the rods and cones connect with a set of retinal cells called bipolar cells, which in turn connect with retinal ganglion cells whose fibres form the optic nerve which links the eye to the visual area of the brain. Notice also that the nerve fibres from the peripheral parts of the retina skirt around the foveal area, minimising the amount of obstruction. Nevertheless, objects inside the eye in the foveal area can be revealed by using a piece of card with a pinhole in the centre. It is placed before the eye, looking at a bright field, and oscillated. By blocking off most of the lens, only rays from one direction can reach the retina. By moving the card, the direction of these rays is altered, so shifting the shadows of the objects across the receptors. The internal structures revealed are different from those seen with a moving light since the fovea is free of blood vessels.

Because the retina is inside out, the optic nerve has to pierce the retina to get out of the eye. The place where it leaves is called the "blind spot". If an image is projected on to that spot, it will not be perceived. Yet we don't see our blind spots - we don't experience a gap in our view of the world around us!

So we can see that the eye is a very blunt instrument, compared to a good quality camera. Yet the remarkable thing is that our view of the world is not blurred nor incomplete. Indeed, it is remarkably acute. For example, we can discriminate at least 100,000 different hues: we can see fine details, subtending visual angles as small as 2-5 seconds of arc (tho' the distance between the cones is about 25 seconds of arc where the packing density is highest).

To escape from this paradox, we must stop thinking of the eye as an image forming optical device, and begin thinking of it as a device for converting patterns of light and dark into a code, a set of symbols, that can be manipulated by internal processing mechanisms.

Function

When we discussed the structure of the eye, we noted the existence of several types of cells, namely receptor cells, bipolar cells and ganglion cells.

Apart from cone cells in the fovea which are linked individually to the visual brain, each ganglion cell is stimulated by a number of bipolar cells, and each bipolar cell is stimulated by a number of receptor cells. The extent of the convergence from receptor to ganglion is indicated by noting that one million fibres, in the nerve which connects the eye to the brain, carry information obtained by the action of some 130 million receptor cells. To use a computing metaphor, we might think of this retinal structure as functioning as a "front-end processor", carrying out local computations on behalf of the main processor, i.e., the visual brain. Usually, a front-end processor is used to reduce the computational load on the main processor; in this case, it is probably provided to reduce the number of separate fibres in the optic nerve since a nerve with 150 million fibres would make it difficult to move the eye, due to cable drag. What we are interested in is discovering what kinds of local computations this front-end processor might be carrying out -- always bearing in mind the operations carried out by the artificial system discussed previously. In other words, can we detect neutral mechanisms for computing grey-level descriptions, identifying candidate edge points, and so on? This is our task.

We will begin by examining the properties of the retinal cells since they are the primitive components of the computational mechanisms. We will start with the receptor cells, rods and cones. Neurophysiological evidence from studies of the retina of the mud puppy, a fish that lives in the depths of the silt laden rivers, is helpful. Because it has large retinal cells, the technique of single cell micro-electrode recording can be used to investigate the cells' properties. With this technique, a fine wire probe is placed beside a cell body. A light stimulus of appropriate type is projected on to the eye, the output from the cell is picked up by the micro-electrode, amplified and recorded for interpretation.

What the experiments have shown is that the mud puppy's receptors respond to the luminance of the input pattern in just the way required to create a grey-level description, i.e., the response of the receptor cells is proportional to the light intensity prevailing. So the voltage signal produced by the cell is equivalent to the voltage signal produced by an artificial sensor, at some (x,y) position, before conversion of that voltage into a numerical value and placing that value in the grey-level description.

We saw that the values in the grey-level description are normally organized as a rectangular array since most artificial sensors scan/ sample the image in this way. This is equivalent to a set of cells organized as a 2-dimensional linear array, each producing its own voltage signal for subsequent conversion into a discrete value (as in the case of the

photodiode array sensor referred to previously). In the case of the human eye, the essential difference is that the receptor cells are organized in diagonal arrays, similar to the arrangement of the cells in a bee's honeycomb. This suggests that we must look for an edge element operator with a somewhat different structure from the simple 2 x 2 operator used in the artificial system. Note too that whereas the artificial system built up its grey-level description in a serial fashion by scanning the image from left-to-right and from top-to-bottom, the eye uses a different strategy. Biological visual systems have developed a parallel processing capability. That is, they have chosen to replicate components so that all parts of an input image, and all types of features, can be dealt with at the same time. So we can expect to find multiple copies of the edge element operator in the retinal structure.

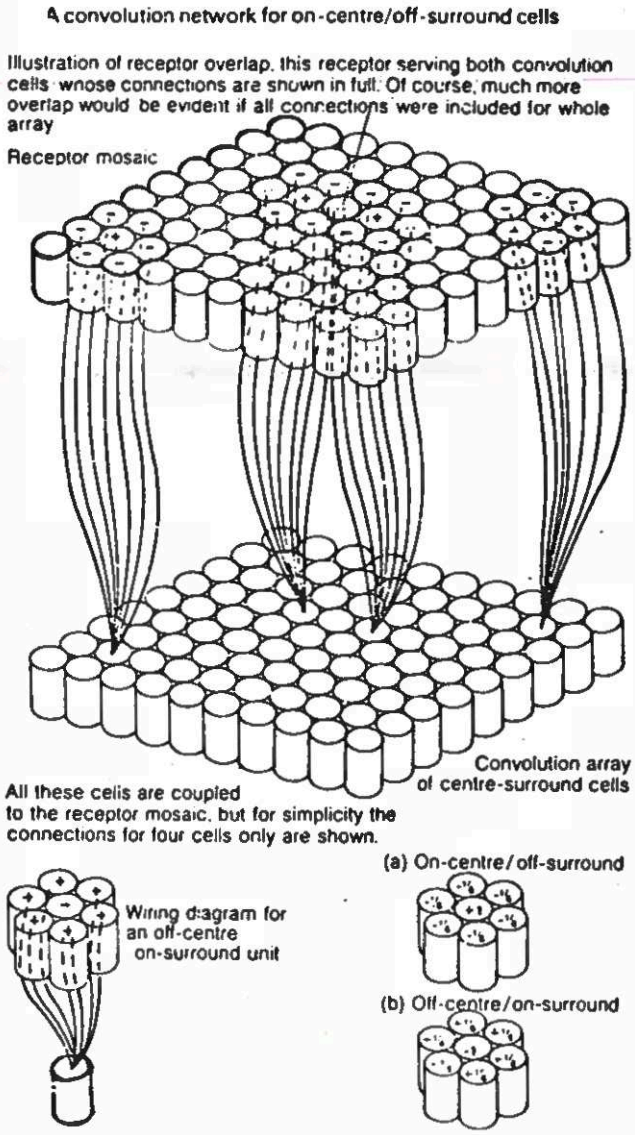
Now we must consider one further property of the retinal cells. Unlike the cells in the artificial system which respond uniformly to light stimulation, i.e., the voltage signal increases as the luminance increases, individual retinal cells in the levels above the receptors i.e. bipolars/ganglia, do not respond in the same way to a given light intensity. Some increase their activity when stimulated with light; others decrease their activity. Cells which increase their activity are called excitatory cells; those which reduce their activity are known as inhibitory cells. So we are looking for an edge element operator which is constructed out of some combination of excitatory and inhibitory cells.

Again we get a clue by considering the neurophysiological evidence. Recordings from ganglion cells in the eye have shown that each ganglion cell is stimulated by a group of receptor cells. Known as the ganglion cell's receptive field, it is roughly circular in form but it has a central area which differs in sensitivity from the surrounding annulus. These receptive fields have been classified into different types, according to their response to light. With a so-called "on-off" cell, when a spot of light falls in the central area, a response is triggered (called an 'on' response), but if the light overlaps the annulus, the ganglion cell's response drops off (called an 'off' response). This effect is called lateral inhibition. The opposite kind of receptive field is also common, in which the surrounding part of the receptive field signals the onset of light ('on' signal), with inhibition of the on-signal when the central area is stimulated. It should be noted that the relatively homogeneous luminance distributions transmitted from the surface of large objects will not stimulate either kind of receptive field since the light will fall on both parts, causing centre and surround activity to cancel each other. So, at the ganglion level there is evidence of the existence of a mechanism for detecting edge elements, located somewhere between the receptor cells and the ganglion-cells.

In the light of what we have learned about retinal cells, let's speculate about the edge element detecting mechanism. Now I am going to introduce you to a new term, convolution. Essentially convolution refers to the process of making an estimate of the goodness-of-fit between a template, which characterizes some kind of feature such as an intensity gradient, and the grey-level representation. So applying the 2 x 2 operator to the grey scale representation in the artificial system was a

convolution process. Now we are carrying out an analogous process, using a different form of operator. (The word 'filter' is often used instead of the word 'operator').

Below, a mosaic of receptor cells is coupled to a convolution array of cells. Each convolution cell extracts from the receptor mosaic a certain limited type of information according to the design of the excitatory and inhibitory connections which feed into it. In any given convolution array, all the cells respond to the same kind of information, but they look at different parts of the input image. As we see, each convolution cell receives inputs from an approximately circular cluster of receptors. The clusters for just four convolution cells are shown.



[left] Wiring diagram for an off-centre/on-surround unit

[right] Weighting diagrams for centre-surround units

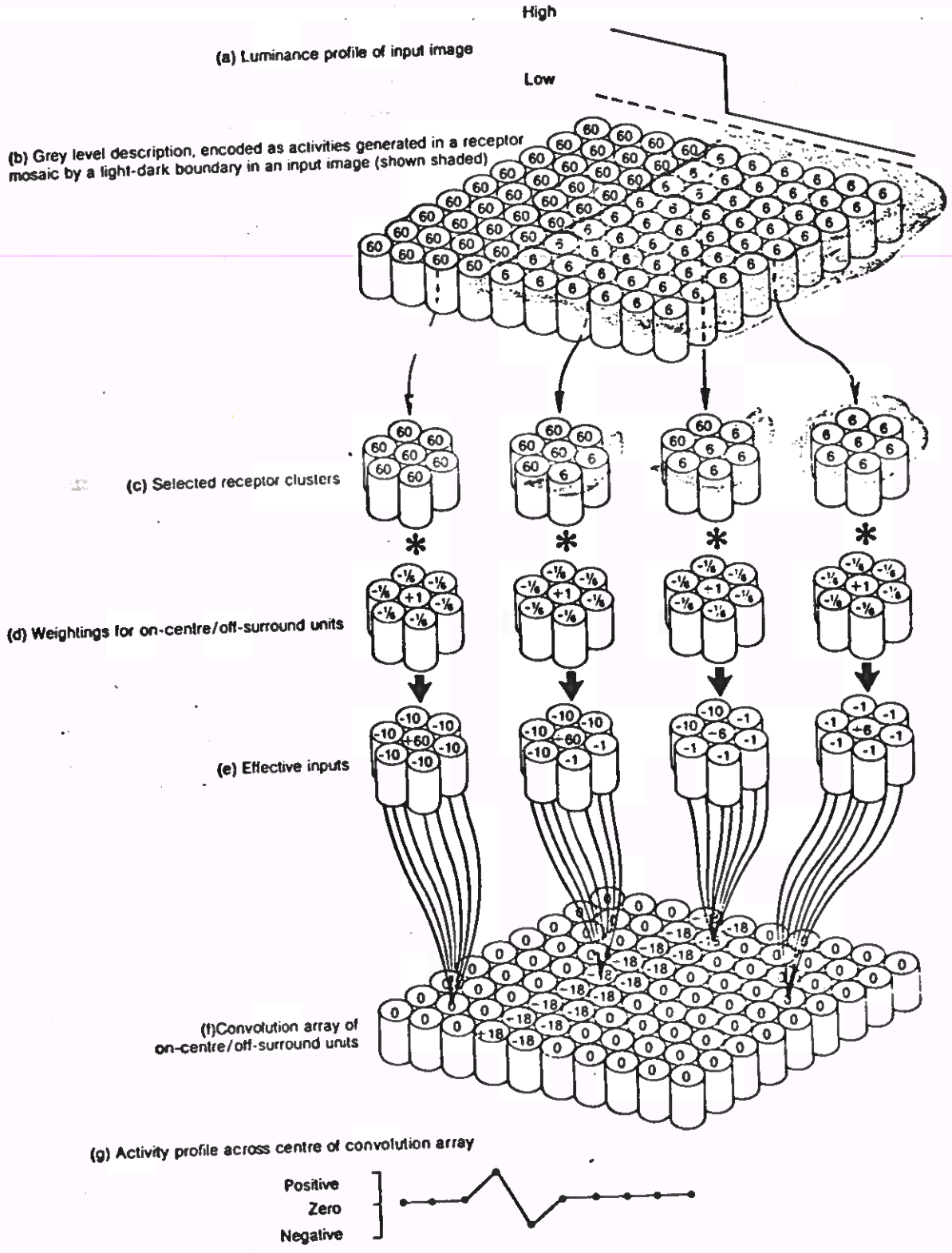
(from Frisby, 1979)

Note that the central receptor in each cluster feeds a convolution cell in the convolution array whose position exactly matches that of the central receptor. The two sheets of cells are thus neatly lined up. In fact, each receptor cell feeds many different convolution cells, but for simplicity only one overlap is shown.

As shown, the central cell in each cluster feeds excitation to its convolution cell, whereas those in the surround feed inhibition, marked by +s and -s respectively. Because the centre-surround connections are antagonistic in this way, i.e. cancel out their respective activities, the convolution cells are usually called on-centre/off-surround units. It is possible for cells to be wired conversely, with the centre feeding inhibition and the surround excitation - these are called off-centre /on-surround units.

The basic objective is to use convolution machinery to detect pixels associated with changes in intensity, i.e., candidate edge elements. This means that the excitatory and inhibitory influences on a cell should add up to zero if the receptor cluster is illuminated uniformly. This can be done by giving each receptor in the cluster a certain weighting in its influence, so that all the receptors are active to the same extent because they are stimulated by an area of even illumination. Then the net influence of all receptors is zero. Suitable weightings for achieving this with our simple centre/surround clusters are +1 or -1 for the centre cells and +1/6 or -1/6 for the surround cells, depending on the type of unit in question.

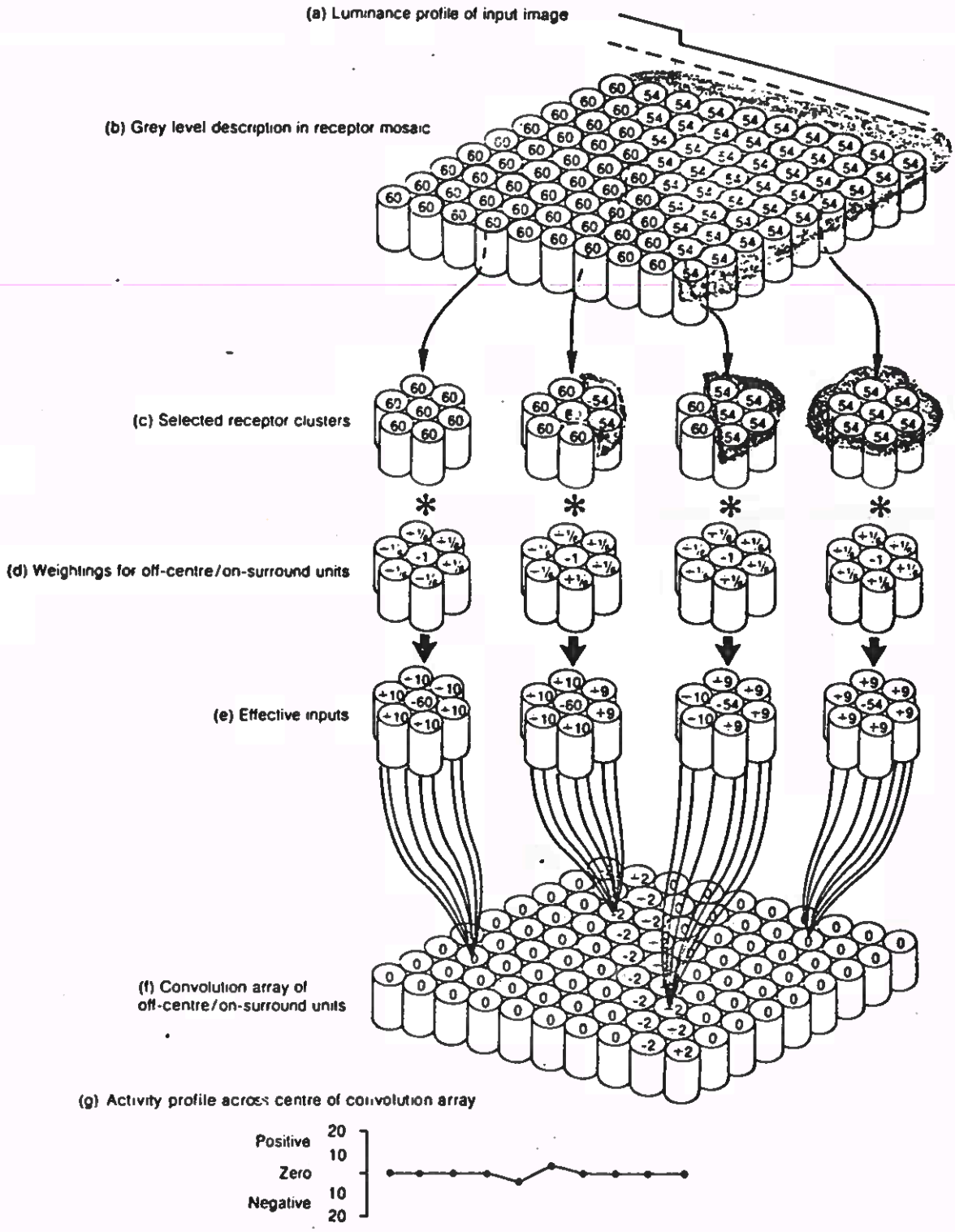
Below, we see an input image containing a steep luminance profile resting on a receptor mosaic. The dark region sets up only weak receptor activity (6 units of activity) whereas the light region induces strong activity (60 units), the whole pattern of numbers constituting a grey-level description.



(from Frisby, 1979)

Now the convolution array looks for pixels at or near the change in luminance, with each convolution cell inspecting one particular region of the receptor mosaic and counting up the excitatory or inhibitory influences coming from this region. Each receptor's activity is multiplied by the appropriate weighting for an on-centre/off surround unit, the results of the multiplication being the effective inputs to the convolution cells. The values in the convolution cells represent the differences between the value of the centre cell and the summed differences of the surrounding cells, and range from 0 to + or - 18 in this example.

Finally, a weak change in luminance, convolved with off-centre/on-surround units is shown below. The procedure is exactly similar to the one which we looked at in detail.

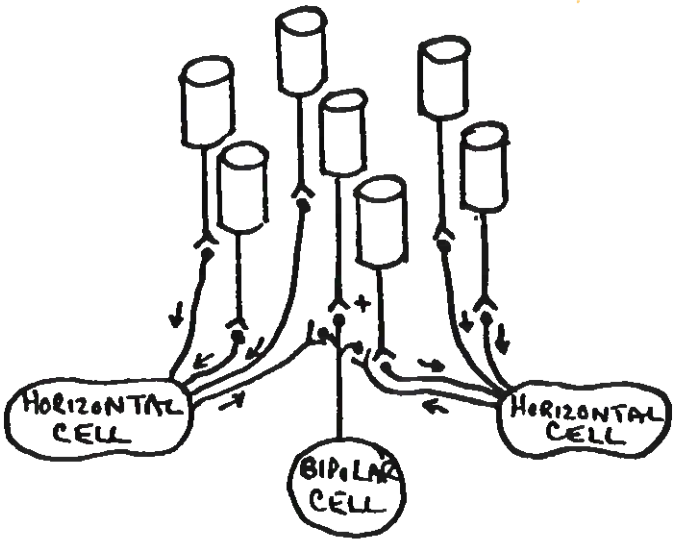


(from Frisby, 1979)

Notice that the orientation of the luminance changes in the two examples are different. In fact, orientation is unimportant for centre-surround units - they are sensitive to changes in luminance irrespective of the orientation. Notice too that it doesn't matter whether off-centre or on-centre units perform the convolution. It just means that the boundary change is represented by a negative-to-positive change going from light-to-dark when an off-centre cell is used, instead of the positive-to-negative change produced by the on-centre unit. The reason for both types is clear, given our knowledge of nerve cells, i.e., a cell can be active or inactive, but it cannot be "negatively active".

Notice that the weak luminance change has produced a weak signal in the convolution array. It is more likely that such a small change represents a difference in illumination rather than a difference in reflection, and so it ought to be disregarded. These small values are filtered out by comparing them against a threshold value.

Turning once again to the neurophysiology, it seems likely that the bipolar cells are responsible for making the centre-surround edge element measurements. Below, we see a highly schematic and simplified wiring diagram of an on-centre bipolar. It receives excitation from a central receptor via a synapse. Other receptors surrounding this central receptor feed inhibition to the bipolar, but not directly. Instead, they feed into horizontal cells which then proceed to inhibit the bipolar. The horizontal cells look like the ideal mechanism for providing the required weighting of surround receptor cells.

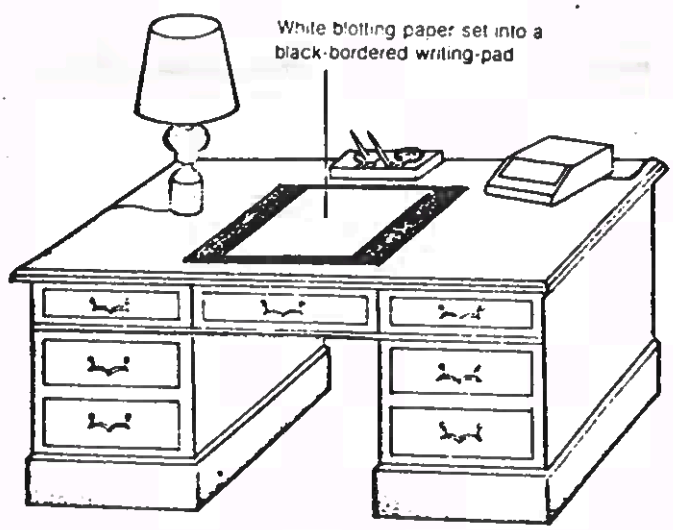


Notice that instead of being stimulated by a single receptor, a bipolar is more usually stimulated by several receptors. This makes the bipolar more sensitive, albeit at the price of loss of accuracy because it would not be able to respond to a spot of light being moved around its central field: it would always judge the spot to be in the same place.

SEEING LIGHTNESS AND BRIGHTNESS

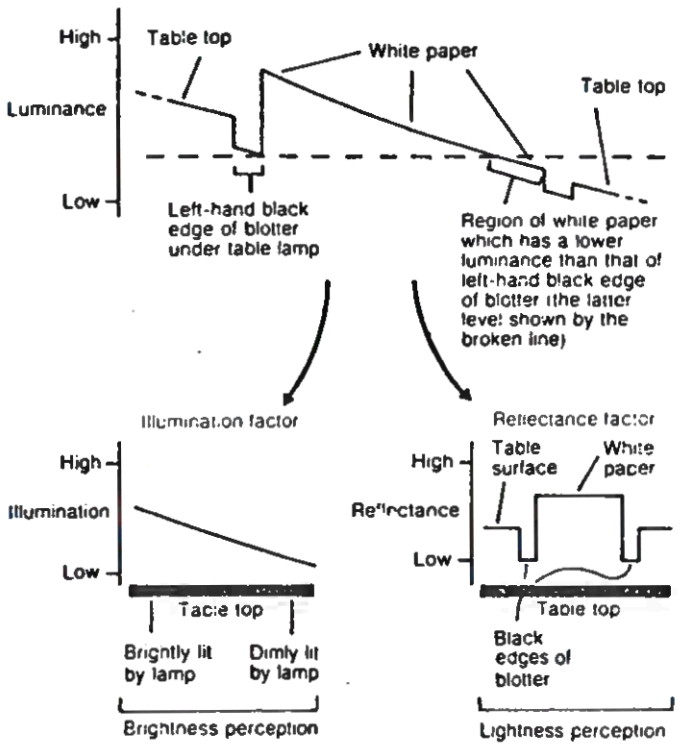
In the artificial seeing system we were dealing with a situation where the illumination was distributed relatively evenly across the surfaces of the object in the scene. But that was a simplification: more usually, the surfaces of objects are not illuminated evenly. More often than not, the light rays from the source will strike each surface of each object at some angle from the normal, producing a different illumination gradient in each case. Yet, on the whole, we perceive each surface as uniformly (i.e. evenly) lit. This is a paradox: if the amount of light falling on a surface decreases with its distance from the source, the surface should appear darker with increasing distance. Can we discover a mechanism in the visual system that delivers a uniform output when supplied with a non-uniform input? This is our next task.

Let's start by making the problem more concrete. Imagine that you are standing in a darkened room, looking down at a desk top whose surface is illuminated by a lamp which is sitting at one side. A large black-edged blotter, holding a sheet of blotting paper, is lying on the centre of the desk top, as shown below:



(from Frisby, 1979)

You will have no difficulty seeing that the edges of the blotter are black and that the paper is white. Yet, because the lamp is positioned at the side of the blotter, the black edge lying directly under it reflects more light to the eye than the far edge of the paper which still appears white; this situation is represented schematically in terms of the likely luminance profile across the desk top, as shown below:



(from Frisby, 1979)

This profile suggests that the blackness/whiteness of a surface does NOT depend simply upon the amount of light entering the eye from a surface, otherwise the physically black surface under the lamp would appear WHITER than the white surface distant from the lamp.

Given that the reflectance of the black border is relatively constant and that its appearance (perceived as lightness) is the same despite the variation in illumination, this suggests that the human visual system is able to factor out the illumination variation (perceived as brightness).

Again, we can represent the luminance profile, and its breakdown into an illumination factor and a reflectance factor, as shown above.

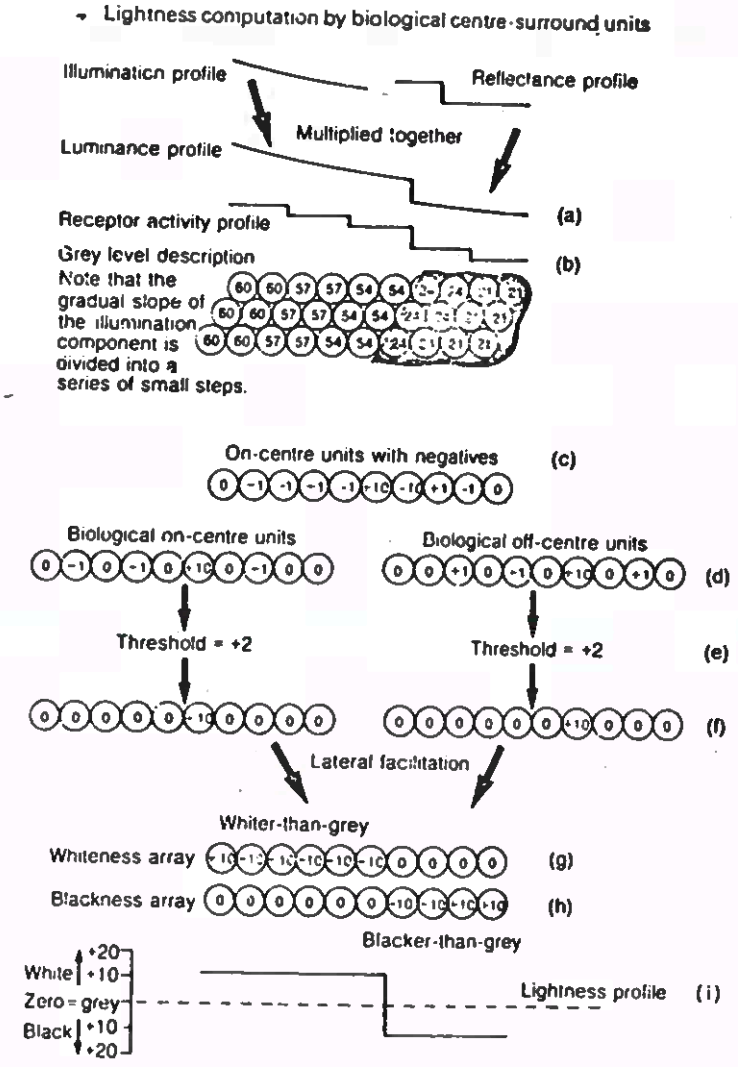
The question is how might the visual system do this? One possible

answer is that it takes advantage of the fact that variations in illumination are relatively gradual, whereas variations in reflectance are rather abrupt. So the unwanted illumination component can be eliminated as follows:

First, detect edges by the convolution process described previously. The effect of the filtering will be to eliminate gradual luminance transitions while preserving sudden ones.

Second, build up the required lightness profile by reconstituting between edges. This amounts to 'joining up' areas between above-threshold edges, giving these areas lightness values determined by the size of the luminance differences forming the edges. This process is harder to achieve but is essentially just the opposite of the original centre-surround convolution. Because of this, it is often called deconvolution. Deconvolution can be performed by arrays of units which facilitate each other adjacently. Whereas the centre-surround convolutions had as their key feature the antagonistic influences of excitation and inhibition, the final step in lightness computation uses excitation only, so that activity can spread out from the edge.

Consider, for example, a luminance profile made up of a gradual illumination change superimposed upon a sudden reflectance change; as shown below:



(from Frisby, 1979)

The objective of lightness computation is to extract the reflectance profile from this ambiguous input.
The first step is the grey-level description, in the usual form of levels of activity in the receptor mosaic, as shown in (b). Its numbers show that the illumination component appears in the form of a set of small steps, of 60/57, 57/54, and 24/21. On the other hand, the reflectance step is much greater, 54/24.

Next, the grey-level description is convolved with on-centre units. Recollect that an on-centre unit is of the form:

For the fragment of grey-level description given, this produces the output shown in (c).

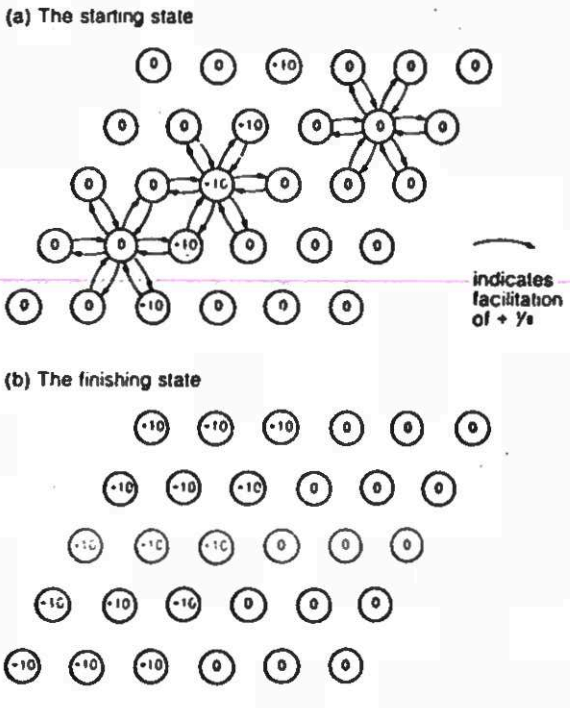
But 'biological' centre-surround units cannot signal negative values, so any negative cell is set to zero. The convolutions for our fragment (without negatives) are shown in (d). The positive numbers appear as positive numbers in the on-centre convolution whereas the negative numbers appear as positive numbers in the off-centre convolutions.

The next step is to apply a threshold. In this example, it is set to +2, leaving just the large edge measurements of +10 as the only ones appearing in each convolution array, as shown in (f).

Once the threshold has been applied, the final step is to build up the required lightness profile by extending the activity outwards from the above-threshold edges. This operation is done in two new sets of arrays, termed the whiteness and blackness arrays, as shown in (g) and (h). The whiteness array shows the results of extending out from the edge recorded by the on-centre units, and the blackness array does likewise for the off-centre units. Somehow, activity in the whiteness array is not allowed to spread in the wrong direction, across the white-black border, and vice versa. This might be achieved by coupling together each white-black pair of cells dealing with the same part of the grey-level description, so that whichever cell is more active "wins out" and inhibits the other one to zero level. So any facilitation passed across the edge within either array would never exceed the value of the inhibitory opponent cell.

The last task is to explain how the spread of activity necessary for deconvolution is achieved within the whiteness and blackness arrays by the process of lateral facilitation. The network of connections for doing this is shown below. The starting state is shown in (a), and the finishing state in (b).

Reconstitution by deconvolution



Deconvolution by lateral facilitation
(a) The starting state Connections for just a few cells are shown but all cells are in fact connected up identically.
(b) The finishing state All connections have been removed here for simplicity.

(from Frisby, 1979)

This could be a blackness or whiteness array. Each cell both influences and is influenced by its neighbours. In the example, each cell excites its neighbours by 1/6 of its own activity level. So each cell is helping its neighbour and being helped out by them. This process goes on and on until a steady state is achieved by the network.

Consider (above (a)) the cell, second from the bottom and second from the left. It starts from zero, the value assigned after thresholding. So it offers no excitation to its neighbours. But it receives excitation from two neighbours which started from +10 because they are "on" an edge. Since +10 x 1/6 = 1.67, the vital facilitation received by this unit is 3.34. Now it can facilitate its neighbours, and can in turn be facilitated by them until the whole network arrives at a steady state, as shown in (b). The mechanism postulated above is made more plausible since neuro-physiological evidence suggests that receptors can feed two bipolars simultaneously. One bipolar of each pair might be an on-centre unit, the

other an off-centre one. Certainly about equal numbers of bipolars of each type are found in the mud puppy's retinal structure. The existence of this pair of bipolars fits with the notions of the whiteness and blackness channels.

If the bipolars are the site of the first step in the lightness computation i.e. edge detection, where are the sites of the next two processes, namely thresholding and deconvolution? It has been suggested that the bipolars are well suited to operate in a threshold manner, which would mean that they respond only if the edge with which they are dealing is sufficiently prominent. How their threshold for responding is adjusted, as it must be to cope with variations in the overall level of illumination, is not known, but must be set somehow by horizontal cells or amacrine cells.

The final question is deconvolution. Marr has suggested that the deconvolution operation is carried out at the bipolar-ganglion cell junction, and is initiated by the lateral connections provided by amacrine cells. The general idea is that there are two sets of ganglion cells, one set carrying the whiter-than-grey lightness information and another set dealing with the blacker-than-grey. Each pathway would be fed by bipolars of matching type, and the close coupling between them might be performed by yet other types of amacrine cells. Of course this proposal is speculative. The conventional view is that ganglion cells are edge detecting units, much as described above for the bipolars, i.e., they are units which help to detect contrast changes despite variations in general luminance. So we must keep our minds open on this issue.

FEATURE DETECTORS : ARTIFICIAL SYSTEM

So far, we have considered how an artificial eye might process visual data to help us understand some of the kinds of local computations which the human visual system might make early on in the processing (in effect, at retinal level). Now, we will consider the next step in this processing hierarchy, namely, describing an object in an image in terms of its edges. As before, we will consider how an artificial system might find edges in an image as an aid to understanding how this might be done in the human visual system.

Recollect that we discussed previously a two-dimensional differencing operator (Roberts' Cross operator) which was applied to the luminance values in the grey level description to yield a new numerical description which characterises the contrast in the image (sometimes referred to as a differential description). Suppose a program equipped with this operator is being applied to the domain of regular polyhedral objects (our current assumption as it happens). This choice of objects means that the edges in the image will be straight edges. These straight edges ought, therefore, to be represented as rowns or columns of high values in the differential description. Given perfect input data, including perfect conversion to digital form, all high values would signal the presence of some significant discontinuity in the physical world. However, since we know that the acquisition process is not perfect, some of the points will have high values due to noise in the system. That is why these high value points are deemed to be candidate edge points. In other words, although there is a high degree of probability that these points denote the edges of objects in the scene represented by the image data, not all of them will be associated with edges. So how can the system distinguish actual edge points from noise points? The answer is that it uses information about local relationships between points.

Given that the edges of objects will be represented by columns and rows of high values in the differential description, what criteria should we apply? In the case of straight edges, there are three criteria:

(i) Similarity : this refers to the similarity of the individual edge elements, i.e. the candidate points.

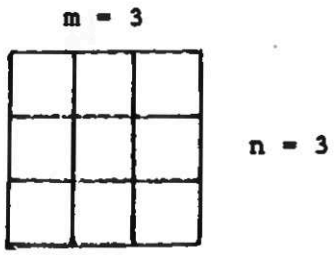
(ii) Adjacency : this refers to the proximity of the individual edge elements. At this level in the analysis, the criterion of adjacency is usually taken as location in neighbouring cells.

(iii) Collinearity : this refers to the spatial relationships of the individual edge elements. To be collinear, the points must lie on or closely approximate to a straight line.

There are various ways of applying these criteria in a program. The one which we will consider is the use of "templates". A template incorporates the criteria given above in its structure. If a program's task is to locate edges in the differential representation, we would wish to equip it with a set of edge templates. These would take the form of rectangular arrays of

cells, m pixels by n pixels, where m is a minimum of 3 pixels and n is a minimum of 3 pixels.

Suppose the task is to detect the presence of high contrast vertical edge segments in the differential description. The system would use a vertical edge template for this task. This is shown below.



Suppose that the differential values lie in the range 0 (no change) to 15 (max. change). It would make sense to parameterize the edge template as follows:

5	10	5
5	10	5
5	10	5

This template will detect vertical <u>boundary</u> edges between background and surfaces of the object. This is done as follows. The system compares the template with the differential representation, searching for places where it might match, using the following match rule:

> Given that the high value column in the 3 x 3 template is assigned the value 10, and the low value columns the value of 5, add 1 to the value of the match score for each cell in the differential representation which corresponds spatially to a <u>high</u> value cell in the template <u>and</u> has a luminance value of 10 or more, and add 1 to the value of the match score for each cell in the differential representation which corresponds spatially to a <u>low</u> value cell in the template <u>and</u> has a luminance value of 5 or less. The <u>entire</u> template is said to match at any position for which the total value of the match is 6 or more.

This will return evidence of the existence of vertical edge segments (3 pixels long) in the differential representation. These can be stored for subsequent processing by recording the co-ordinate values of their <u>end points</u> and their orientations (90^0 in this case). To locate evidence of horizontal and diagonal boundary edge segments, the same process is repeated again and again, using templates whose orientation varies from vertical through intermediate inclinations to the horizontal, i.e. in the range 90^0 to 0^0 in steps.

So far, we have discussed detecting boundary (i.e. sharp, high contrast) edges. Suppose, however, that we also want to detect the

presence of internal edges (where an object's surfaces intersect). In the differential representation, these are characterised by lower contrast (smaller differences) and increased spatial extent (spread over 2 or more pixels width). Accordingly, a new set of templates is required. In effect, these are scaled up versions of the boundary edge detector - say 6 x 6 pixels, with a pair of high value columns flanked by pairs of low value columns. As before, these templates are compared with the values in the differential representation to yield evidence of internal edge segments. These are recorded in similar fashion to boundary edge segments.

Just as we referred to the high value points in the differential representation as candidate edge points, so we ought to regard the edge segments identified by the template matching process as candidate edge segments. These segments have to be combined to form longer edge segments corresponding to entire boundary or internal edges. While the majority of the short segments will be conflated to form these longer segments, some will be rejected as being spurious segments (due to noise in the system).

Recollect that templates incorporate rules for grouping together candidate points. Now, we need to apply similar grouping rules to the candidate edge segments to yield, in due course, an edge description where each edge in the description corresponds to a physical edge in the scene. The grouping rules are as follows:

> If the end point of one segment is adjacent to (e.g. above, below, to-the-left of, to-the-right of) the end point of another segment, and
> If the orientation of the first is the same as the orientation of the other, and
> If the combined segments (combined points) are collinear
> Then link the segments (to form a larger segment).
>
> Junction (corners) are detected as follows:
>
> If the end points or two (or more) segments are adjacent, and
> If the orientation of one is different from the other(s)
> Then combine the segments (to form a junction).

To apply these rules to the candidate edge segments, numerical values have to be assigned to the parameters "adjacent" and "orientation". Also, an error value has to be assigned to the procedure which determines whether or not two segments are collinear. In practice, since, for example, there will be small gaps due to noise, these rules must be applied more than once. Usually, the parameter values will be altered between successive applications, for example, to enable two adjacent edge segments separated by two pixels (which were not linked first time around) to be joined together to form a longer edge segment, and so on. Finally, any candidate edge segments which do not have connections to other edge segments at both ends are removed (these are 'dangling' edges, caused by noise).

At first sight, this method is attractive but the major difficulty in the context of an artificial visual system implemented on a machine with a single processor is that different templates and/or different match rules are needed for vertical edges,

horizontal edges, bright edges, dim edges, sharp edges, fuzzy edges and edges at arbitrary orientations. In other words, the process of using templates is computationally costly. Special purpose parallel computers which can carry out these operations concurrently over the whole image have been under development for some years. In due course, they are likely to replace single processor machines, at least for handling low level processing

FEATURE DETECTORS : HUMAN SYSTEM

Local Feature detectors

We turn now to look at the evidence for the local computation of edge descriptions. Much of our knowledge comes from studies with animals, from which one generalizes (with care) to man.

Apart from lateral inhibition, up until 1959 we knew very little about the coding in the visual system. In that year, Lettvin, Maturana, McCulloch and Pitts announced that the frog's retina contained four kinds of ganglion cells, described as:

(i) Sustained contrast detectors, which indicate static edges of high brightness gradient.

(ii) Moving edge detectors, which signal moving edges of abrupt brightness change.

(iii) Net dimming detectors, which respond to a sudden reduction of illumination (approach of a predator, perhaps) and

(iv) Net convexity detectors, which respond when a small bright spot enters the visual field (insect in view, perhaps).

These results were obtained by micro-electrode recording from the ganglion cells in the frog's eye, when the eye was stimulated with an appropriate pattern of light and dark.

Later research by Hubel and Wiesel, working with cats and monkeys, has failed to disclose retinal cells with such discriminatory properties as those found in the frog's retina. This is not surprising because the frog, unlike higher animals, does not have a visual processing area in its brain. However, Hubel and Wiesel showed that cells in the monkey's brain are sensitive to different types of visual features. In fact, they identified two major classes of brain cell, namely the simple cell and the complex cell.

Before we consider the properties of different types of simple cells (we will not consider complex cells since their role is still not adequately understood), we will examine some interesting evidence about the way they are organized in the visual area of the monkey's brain. What Hubel and Wiesel have shown is that the visual cortex is rather like a bee's honeycomb: it is divided into tiny segments, each extending from the surface of the cortex (the grey matter) vertically down into the white matter, deep within the hemisphere. Each segment represents a processing sub-unit, called a hypercolumn by Hubel and Wiesel. Each hypercolumn's total area is approximately 0.5 - 1 mm square at the cortical surface, and about 3 - 4mm thick (approximately the full thickness of the cortex); it contains tens of thousands of cells, perhaps up to a quarter of a million. The job of all the cells in a particular hypercolumn is to inspect jointly a particular region of the retina, a region called the hyperfield. While hyperfields overlap to some degree, essentially each hypercolumn is

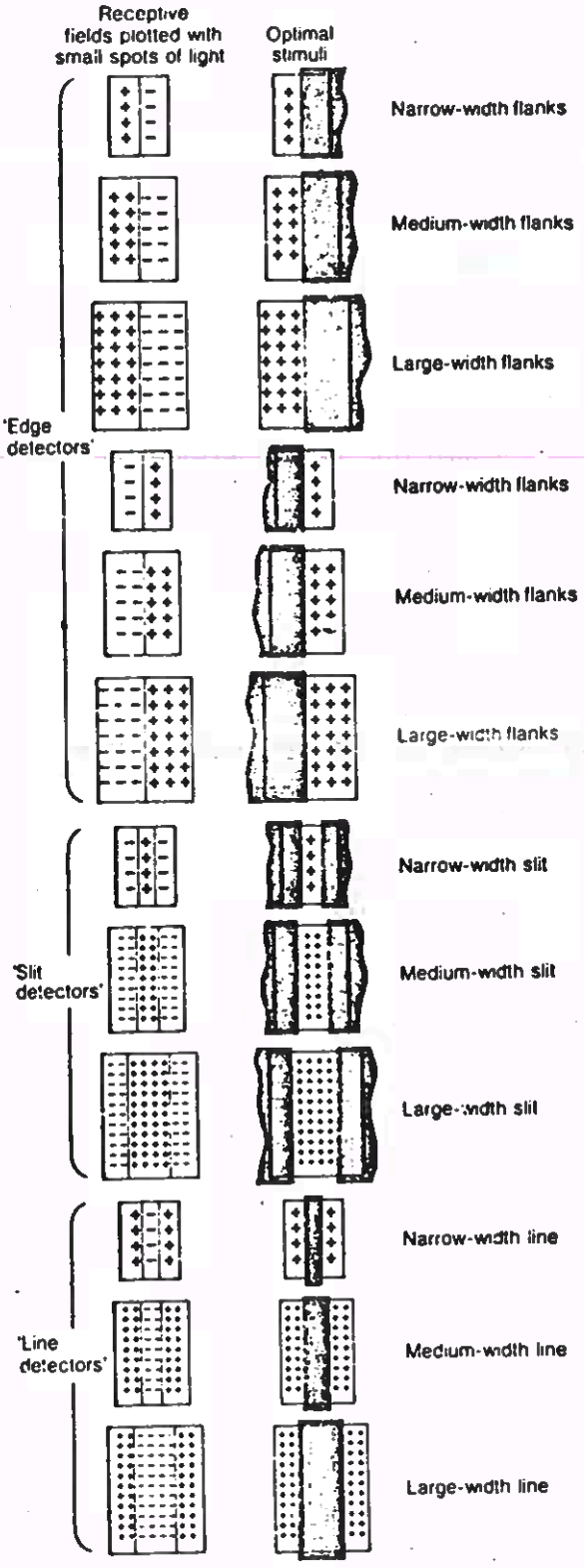
concerned with just one region of the input image. Thus the hypercolumns all "chatter" simultaneously about the features they are "seeing" in their own restricted domains, and it is the job of later processing mechanisms to sort out from this feature description what objects are present in a scene.

While distributed evenly over the cortex the size of the hypercolumns concerned with the central retina differs from the size of those dealing with the periphery. Hypercolumns handling peripheral areas of the retina have large hyperfields and hence can only carry out a crude feature analysis. Central hypercolumns, on the other hand, have smaller hyperfields, so they can engage in much finer analysis. But this is what we would expect, given our knowledge of the retinal mappings from receptors to ganglion cells. Note that because central hypercolumns have smaller fields, more are needed to cover a given area of the retinal surface. This fact indicates that the spatial mapping from retina to cortex will be significantly distorted, with the periphery of the visual field compressed relative to the centre.

When a micro-electrode is driven down through a hypercolumn, besides the fact that they all have their receptive fields in the same general region of the retina, all cells share a very important property, irrespective of their type: they are all maximally excited by stimuli with the same orientation.

The simple cell's distinguishing characteristic is that its receptive field can be divided into excitatory and inhibitory sub-regions, using stationary stimulation. So if a spot of light is flashed on certain regions of the simple cell's receptive field, the cell becomes excited and emits a burst of impulses. Equally, if flashed on other regions of the simple cell's receptive field, the cell becomes inhibited and stops emitting pulses.

What are the shapes of these regions? This question can be answered by exploring the effects of flashing small spots of light all over the cell's field, noting each time the effect of the flash on the cell. By recording excitation and inhibition with plus-signs and minus-signs on a paper representing the region of the retina covered by the receptive field, typical field maps of simple cells can be obtained. Some are shown below. By careful examination, we can group them into different types called edge detectors, slit detectors and line detectors.



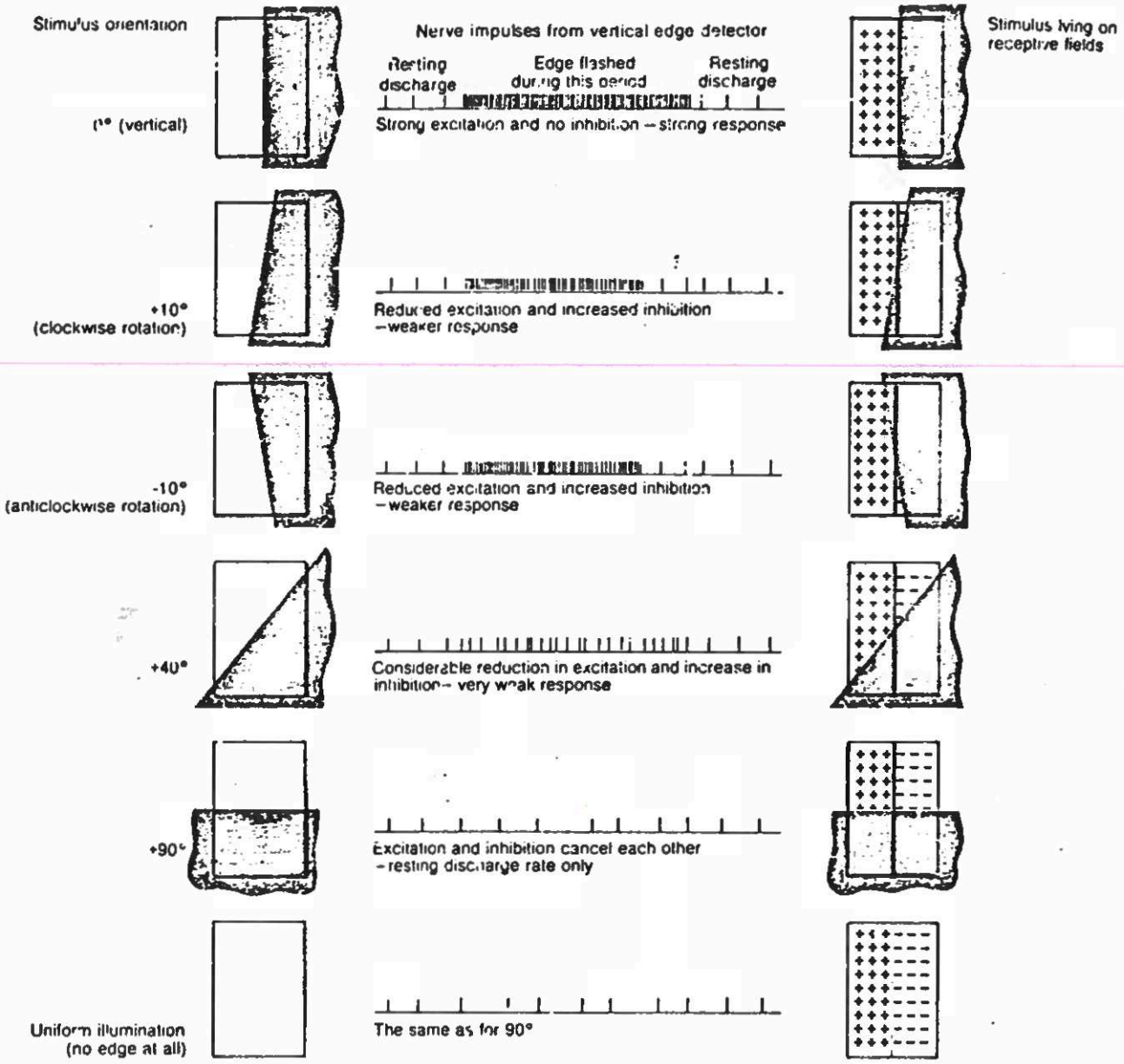
(from Frisby, 1979)

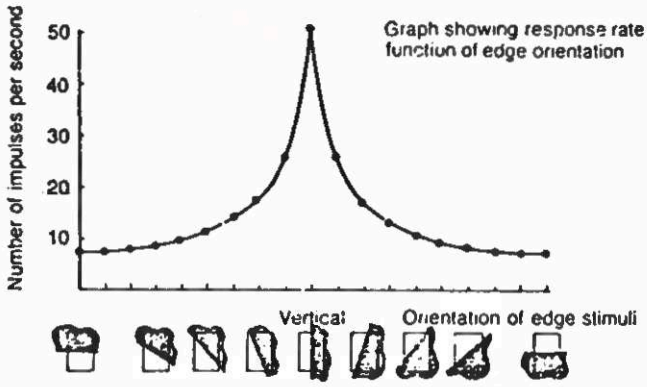
Consider first the receptive field of the so-called edge detecting simple cell. Note that each field is divided into two sub-regions, one excitatory and one inhibitory. The boundary between these sub-regions has an orientation which defines the orientation tuning of the cell in question. Notice that all edge detectors are vertically tuned, i.e. they respond maximally to vertical edges. Indeed, all the field maps shown are vertically tuned. This is because all the field maps come from cells within a single hypercolumn. Remember that cells within any one hypercolumn share the same orientation tuning, with different hypercolumns differing in the orientation to which they are tuned.

The term "slit" (which describes a stimulation of a white line on a dark surround) is a little odd, but refers to the stimulation arrangement used by Hubel and Wiesel - shining light through a slit. The term "line" has become a customary one for a dark line on a light surround, an unfortunate usage because all the stimuli shown are line stimuli of a sort, and not just those termed "line". Sometimes, however, the slit and line stimuli are called light and dark 'bars' respectively.

Now we can undertstand why these cells have so often been dubbed feature detectors. That is, it has been commonly assumed that because each cell has as optimal stimulus one or other line features, each cell must be a signalling device for saying whether this feature is present on the patch of the retina inspected by the column of cells as a whole.

But what happens if a non-vertical stimulus falls on a vertically oriented receptive field? The answer is that the response diminishes by the amount the stimulus diverges from the vertical. This is illustrated next, for a left-right light-dark edge.





(from Frisby, 1979)

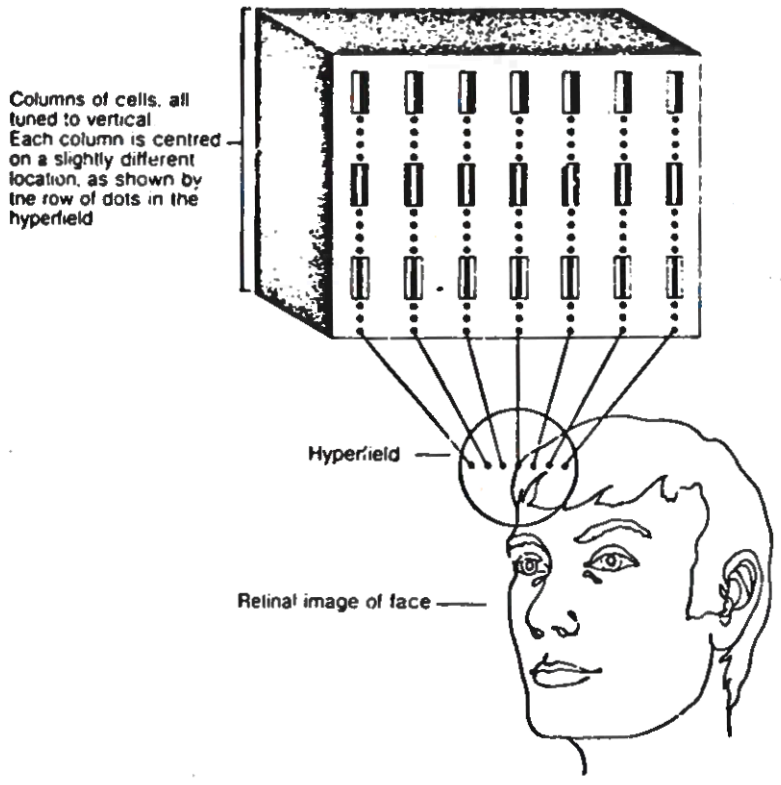
The effect of stimulus orientation, for many more edge stimuli than shown in the diagram, is illustrated in the graph. We can see why the edge detector exhibits this vertical tuning effect. As the stimulus edge is rotated, darkness falls on some of the excitatory zone, and at the same time light falls on the inhibitory zone. By the time the edge is horizontal, the cell receives equal amounts of excitation and inhibition, so its firing rate is reduced to the spontaneous firing level.

It is best to think of the excitatory and inhibitory zones of each cell's field as carrying equal weight overall. In other words, under conditions of even illumination, there will be an equal balance between them.

Notice that the column of cells are sub-divided . Thus, some cells have "narrow-width slit" optimal stimuli, others "medium- width slit", and so on. The cells we have seen are just a sample of the population in the column, and many more types of field exist covering a wide range of slit-widths, line-widths and widths-of-flanks on either side of an edge. Just why so many different widths are needed will be taken up again later.

We have been concentrating on a single vertically tuned column of cells. But each hypercolumn contains many columns, all similarly tuned but each dealing with a slightly different area of the hyperfield, as shown below:

A slab of vertically tuned columns within a hypercolumn There are many cells in each column. Only a few are shown, as dots. Just three cells in each column are enlarged to show their receptive field types (from top down, edge, slit and line fields).

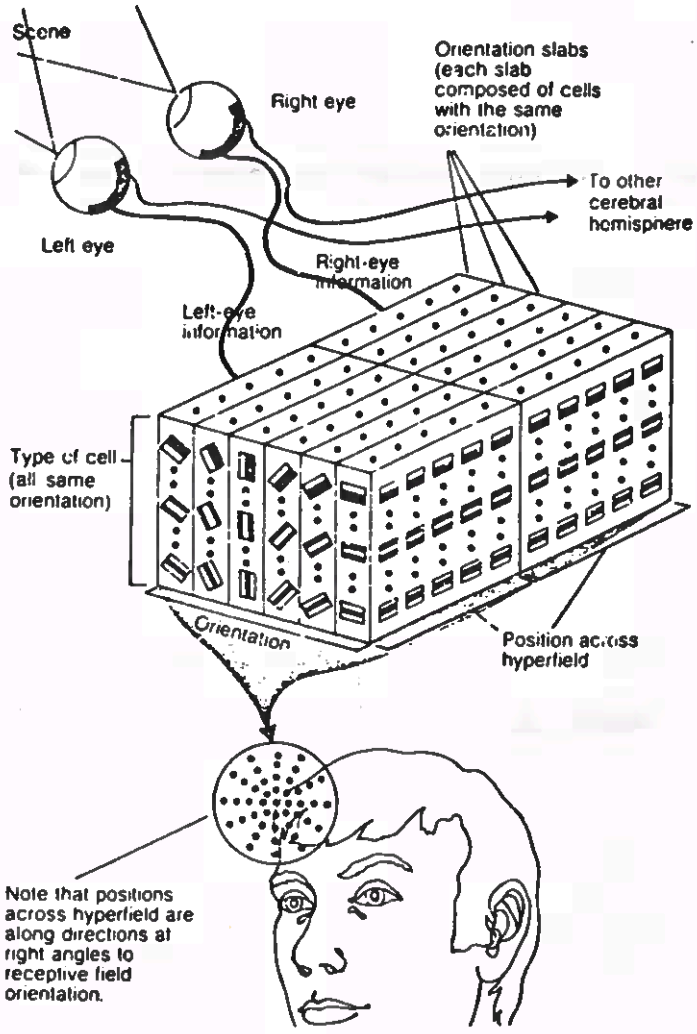


(from Frisby, 1979)

In other words, each vertically tuned column is "centred" on a different spot in the hyperfield so there is a slab of columns covering the whole width of the hyperfield i.e. the columns are slabs like, being positioned side by side. But the receptive fields of neighbouring columns within the slab will overlap to give continuous coverage across the field. Notice that the vertically tuned slab has its columns inspecting points spread out horizontally across the hyperfield, i.e. points are at a right angle to orientation of cells.

Turning now to deal with other orientations, it seems that there are columns for orientations all around the clock, with the tuning of each column differing by about 10 degrees from its nearest "orientation neighbour".

We are now in a position to speculate about the organization of an entire hypercolumn. A model is shown below. In reality, there are many more orientations than this shows - perhaps 18-20 in all to get right around the clock. Remember, too, that there are many different cell types within each column, of which only a few are shown.



(from Frisby, 1979)

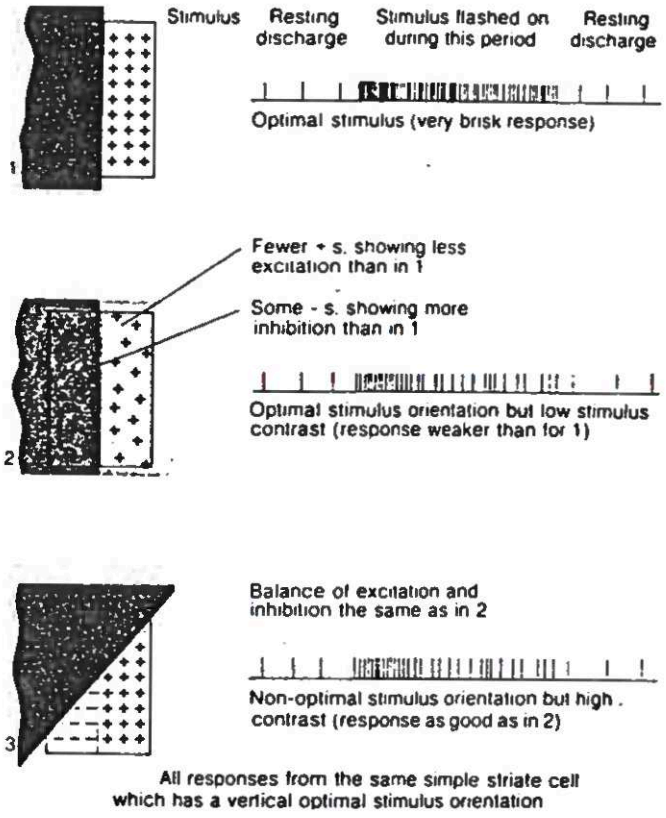
One extra feature is the fact that a hypercolumn really has two halves - a left one and a right one. Although shown as separate, some cells in the cortex are binocularly driven. That is, they respond actively to

optimal stimuli in either eye. Others are preferentially driven from just one eye. So the division into two parts is an oversimplification. None-the-less, the monocular dominance of certain regions of the hypercolumn has been well confirmed by the work of Hubel and Wiesel.

Of course, the hypercolumn structure we have been discussing is hypothetical, but it does fit a great deal of neurophysiological data. We will stick with it for the present.

So much for the structure of the hypercolumn and its components. The next question is now does each hypercolumn examine its own patch of retina (its hyperfield), and arrive at a feature description of the images falling on this patch?

As we saw above, each simple cell seems to signal a particular orientation. Hubel and Wiesel postulated that these cells form the bottom layers of a hierarchy of cells (so far undiscovered) which respond to progressively more and more abstract geometric features. For example, the cells at the next level up might respond to simple geometric patterns such as angles, and so on up to the top of the hierarchy which might respond to stimuli such as particular items of food, particular individuals, and so on. This feature hierarchy theory is often referred to as the "grandmother cell" theory. Persuasive as this might seem to be, we cannot accept Hubel and Wiesel's explanation. Consider a simple cell whose optimal stimulus is a vertical edge: it will respond most strongly when stimulated by a high-contrast black-white edge, as shown below:

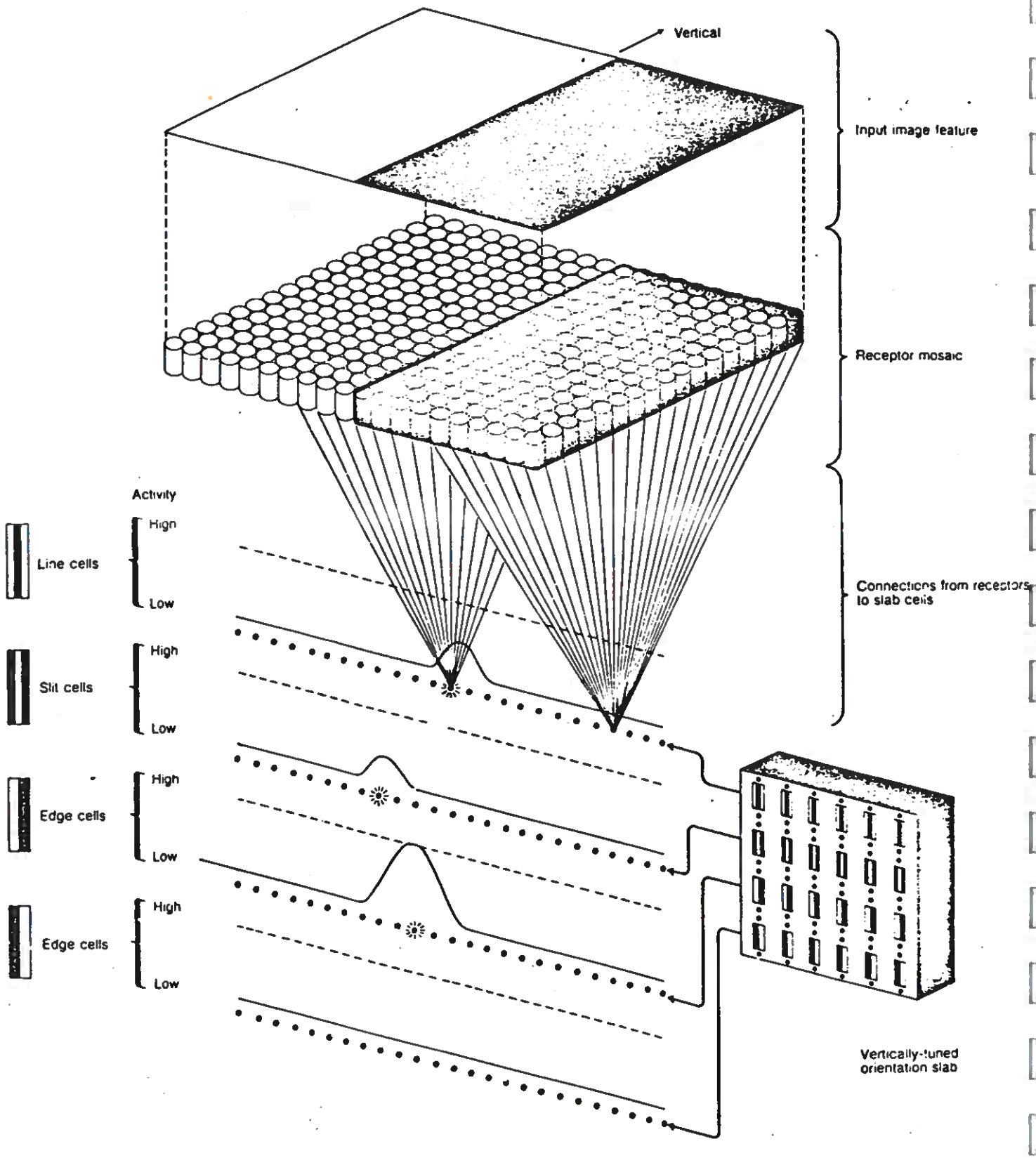


All responses from the same simple striate cell which has a vertical optimal stimulus orientation

(from Frisby, 1979)

If the contrast is reduced by making the black zone a dark grey, and the white zone a light grey, the cell will respond less vigorously. What happens if we stimulate this cell with a high-contrast black-white edge which is rotated a few degrees (± 10) from the optimal vertical orientation? The cell will respond as vigorously to this rotated stimulus as it did to the lower contrast vertically oriented stimulus. So if different stimulus conditions cause a cell to produce the same response, how can it know which condition is occurring? Indeed, if activity in the cell were to be taken simply and directly as the neural representation of a vertical edge, we would be susceptible to some very awkward illusions. We would confuse faint vertical edges with high-contrast just-off vertical ones, a quite unsatisfactory state of affairs which doesn't arise. A similar problem arises in the case of cell type. Once again, more is needed than simply equating edge detector responses with step-like illumination profiles, slit detector responses with light lines on a dark background, line detector responses with dark lines ona light background, and so on.

An image of a vertical edge focussed on a part of the retina which forms the hyperfield of a particular hypercolumn is shown below. As we have seen, the edge feature is represented as a grey level description at the retinal level. Fibres connect the retinal ganglion cells to two cells in the vertically tuned slab of the hypercolumn. For simplicity, only one row of cells in the slab is shown. In practice, fibres would link ganglion cells to the other cells in the hypercolumn.



(from Frisby, 1979)

An activity profile is shown above each row of cells. This activity profile represents the contrast across the edge. The greatest response is given by the edge cell in the third row. But notice that both the line cell (in the top row) and the slit cell (in the second row) also respond to the presence of the step edge. Clearly, therefore, just because a linear slit cell is active does not mean that there is a line-like or slit-like structure in the hyperfield. In other words, not only is the orientation of the feature doubtful, but its nature is as well.

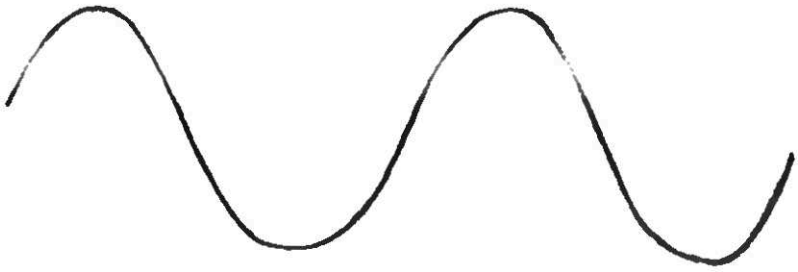
In similar fashion, a line (bar) in the hyperfield not only stimulates line detectors, it also activates slit detectors and edge detectors.

Global feature detectors

In the feature detection approach considered in the last section, simple cells were considered as signalling the presence of particular geometrical features, at particular positions within the pattern of light falling on the retina. The implication is that the combining of local geometric feature information to yield descriptions of global objects must take, place at a higher level, through some kind of grouping of simple cells output.

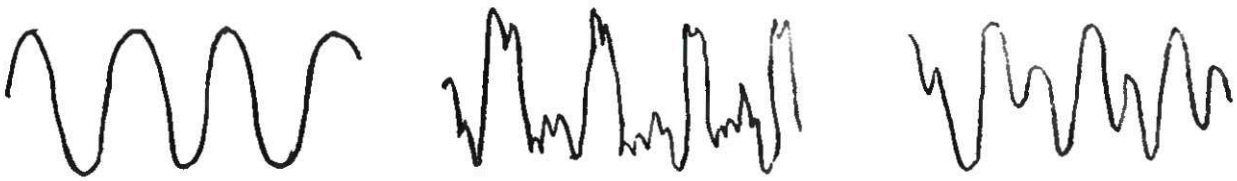
However, there is a quite different explanation of the role of the simple cell, namely, that it carries information about global properties, not local properties, and in particular, that these global properties are spatial frequencies.

What are spatial frequencies? We will answer this question by analogy with sounds, that is temporal patterns of air pressure produced by some kind of instrument. For example, when the sound from a tuning fork is amplified and the signal is displayed on an oscilloscope, the pattern produced is a sine wave, as shown below:



This is the simplest form of sound wave, sometimes called a pure tone. N.B. Warning! The pattern shown above is a graph which takes the form of a transverse wave whereas the sound wave is a longitudinal wave, with particles vibrating in the same direction as that in which the wave is travelling.

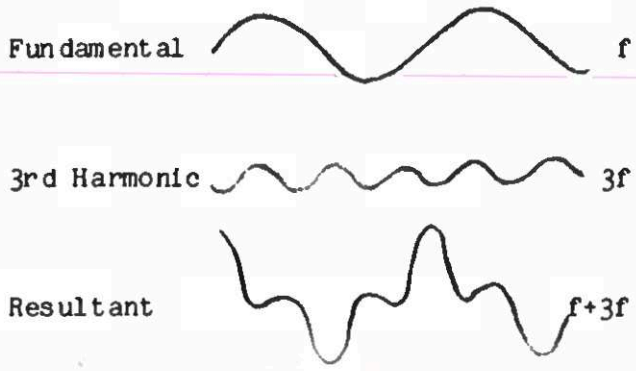
Continuing with the analogy, if an oboe, a French horn and a violin play the same note you have no difficulty in identifying the different instruments. Studing the wave forms on a 'scope enables us to see why the notes can have the same frequency yet sound different. The shapes of the waves are not the same, as shown below:



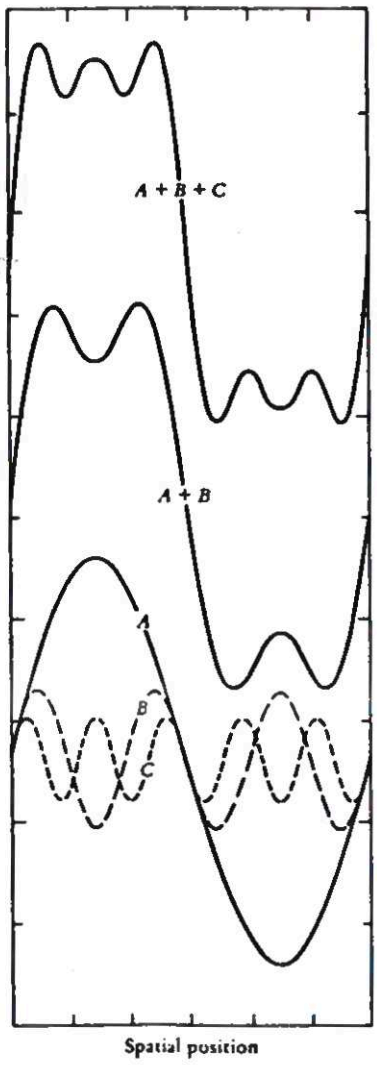
Oboe French Horn Violin

The difference in wave form is responsible for the characteristic sound quality of each instrument. Mathematicians have shown that any shape of wave can be split up into a sine wave at some fundamental frequency and a number of other sine waves at multiples of this frequency which differ in amplitude (harmonics or overtones). This is Fourier Theory.

By adding together the fundamental and one or more higher harmonics, a completely different wave form is obtained. For example, if we add the first and third harmonic, we get the composite wave form shown below:

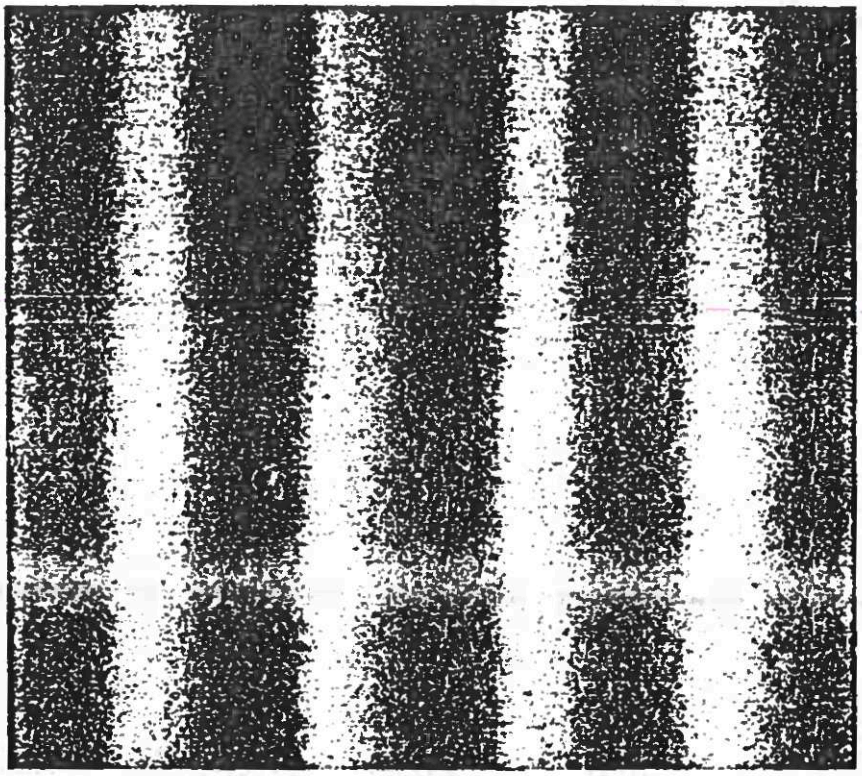


But patterns of light intensity can be described mathematically in the same way as temporal patterns. So, when we turn to consider light instead of sound, we find that the square wave form which represents an edge can be described as the summation of several sine waves of varying amplitude and frequency. For example, a sharp edge can be constructed by adding together more and more sine waves, as shown below:



Just as detecting a particular quality of sound involves detecting the different frequencies of which it is composed, detecting the presence of an edge would involve detecting the different frequencies of which it is composed.

We can control the frequency of light used to stimulate the retina by using a device called a grating shown below:



This is a sinusoidal grating, so its brightness varies sinusoidally across the pattern. This means that the stripes are blurred. A grating is described in terms of frequency , expressed as cycles per degree of visual angle; contrast , expressed as the ratio of maximum to minimum intensity in the pattern, and phase , expressed in degrees, of the pattern relative to a fixed point.

The usual technique is to expose an area of retina to a diffuse field of light, alternating at regular intervals with a sinusoidal grating of the same average light intensity, and to record responses from cells at time of onset or offset of the grating. In this way, the experimenter has precise control over the spatial frequency of the light stimulating the eye: he designs a grating which has the desired characteristics. In practice, the large scale components in the grating are represented by low frequencies, while the fine details are represented by high frequencies.

The point of all this is that physiologists have established that retinal ganglion cells with concentric fields will respond selectively to different spatial frequencies. Also, the smaller a cell's receptive field, the higher the maximum spatial frequency to which it will respond.

At the cortical level, simple cells are tuned to spatial frequency (i.e. each responds to a particular range of frequencies), with each hypercolumn containing cells with a wide range of optimal frequencies.

The response of a cell to a grating is measured by its contrast sensitivity, the reciprocal of the threshold contrast required to obtain a response from a cell. Thus, in the cat's retina, optimum spatial frequencies range from 0.3 to 3 cycles/degree, reflecting its greater visual acuity.

These responses are consistent with the responses to edge, bar and slit stimuli. For example, a cell with a vertical bar as its optimum stimulus will give its maximum response to a vertical grating with wavelength twice the width of the bar, provided the grating is lined up with the boundaries of the excitatory and inhibitory regions.

The unresolved question is why are simple cells selective for spatial frequency? Are these cells detecting global properties in the patterns of light at the retina? Bluntly, the fact that cortical cells respond to spatial frequencies does <u>not</u> prove that the visual system decomposes its input into sinusoidal components any more than the fact that cells are selective for orientation of edges proves that it analyses its input into local geometric features. Indeed, the physiological evidence suggests that cells are not specific in their responses but respond over a broad range of frequencies and have fields of limited size. They are also sensitive to phase, that is, where the peaks and troughs fall in the receptive field. So, we have the same problem as with the ambiguity of the responses of simple cells to features of varying contrast which are set at a range of orientations.

The most promising explanation why each hypercolumn contains cells with different spatial frequency tuning is that multiple spatial frequency tuning is an important characteristic of a system designed to detect edges. It is to this that we turn next.

BUILDING THE PRIMAL SKETCH

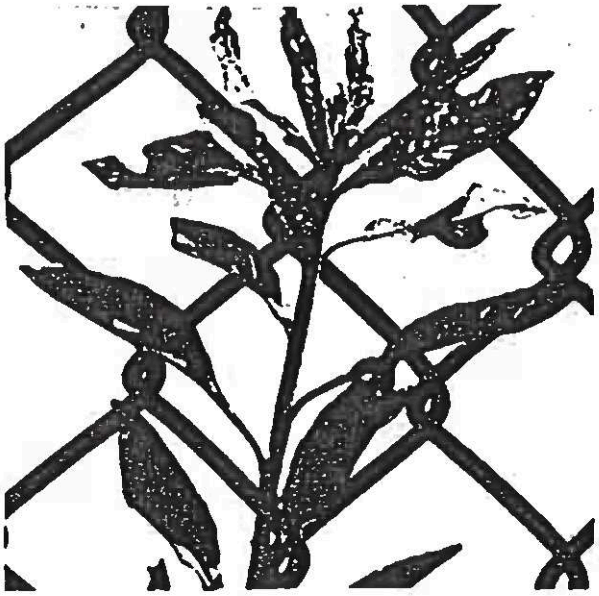
Recently, Marr has argued that the properties of single cells in the visual system can only be understood in the context of a computational theory of vision. He attempted to build such a theory, to explain how the pattern of light at the retina is tranformed into a symbolic description of the environment. The first stage in this transformation process is the construction of a data structure, known as the raw primal sketch. Briefly, the raw primal sketch makes explicit information about the edges and textures of surfaces in an image. The theory specifies how edges can be detected in natural images, and in doing so provides a single explanation for the neurophysiological phenomena which served as the basis for the feature detector and spatial frequency detector theories.

In an earlier section, we examined a simple algorithm for computing differences in light intensity in each region of an image. It made use of a 2x2 mask, and could detect abrupt changes of intensity in an image. It could not , however, handle gradual changes, and it was also susceptible to the effects of noise in the image data. To extract edge information from natural images where the changes in intensity are often very much less abrupt, a much larger mask is required. However, as mask size increases, information about the location of abrupt changes is lost. Hence ,the conclusion that edge detection cannot be done using a mask of uniform size. Rather, the location of intensity changes at differing scales has to be carried out by a number of parallel operations, each using an appropriate mask size.

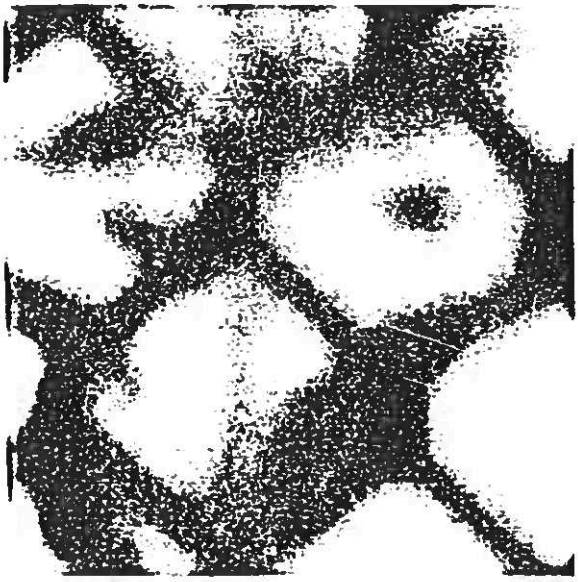
Marr's approach, therefore, is to take the image and transform it into a number of independent representations. In each, there is a different upper limit on the steepness of the gradient present within it. The way in which the steepness of the gradient within a representation is controlled is by blurring the image: the more blurred it is, the shallower the steepest gradient that can be present.

Previously, we discussed the use of smoothing to reduce the effect of noise, by replacing each value in the grey level description with an average of the values of its eight neighbours. Marr's approach is analagous. The essential differences are that he defines areas that are circular rather than rectangular, and that vary in size. Within these circular areas, pixels closer to the centre contribute more to the average through a numerical weighting process in which the values of pixels near the centre cell are multiplied by a higher numerical value than the values of cells nears its periphery. The choice of the values for these weights is not arbitrary: instead, it is done in accordance with the decision that the optimal smoothing function is a Gaussian function (i.e. a bell-shaped statistical distribution). The degree of blurring achieved is determined by the width of the Gaussian distribution, measured in terms of its standard deviation. In practice, Marr filters each image, using two or more Gaussian distributions. In this way, the array of light intensity values making up the grey level representation is replaced by a set of arrays, each containing Gaussian weighted average intensity values. Below, the image (a) has been smoothed using Gaussian distributions with standard deviations of 8 and 4 pixels respectively, giving the more and less blurred images, (b) and (c).

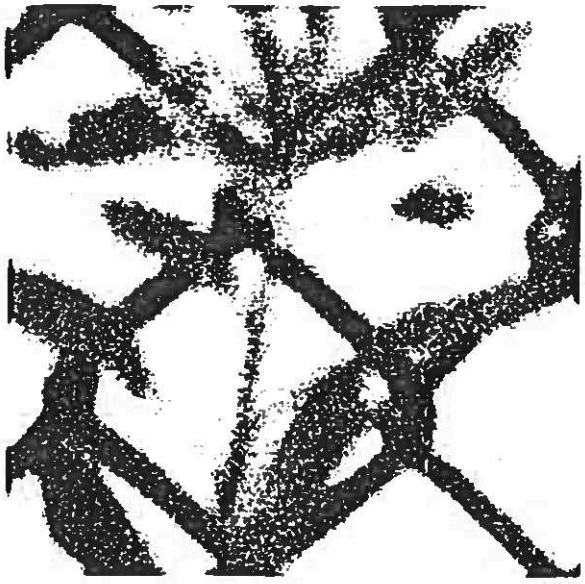
a.



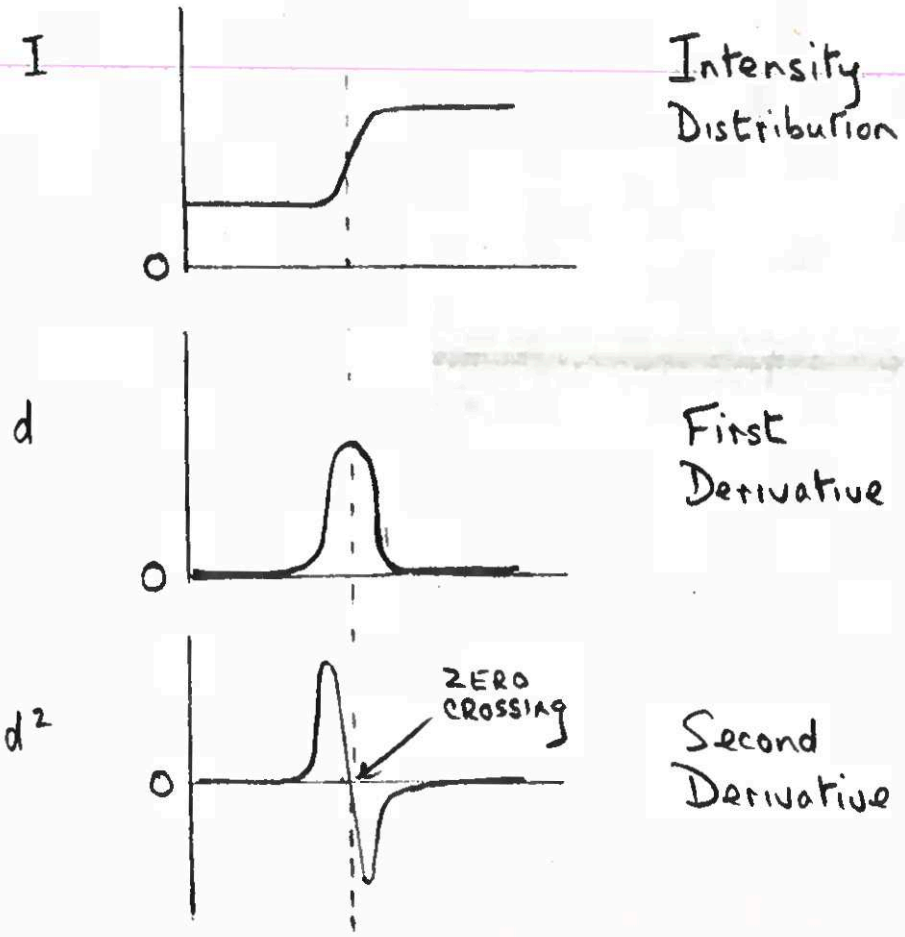
b.



c.



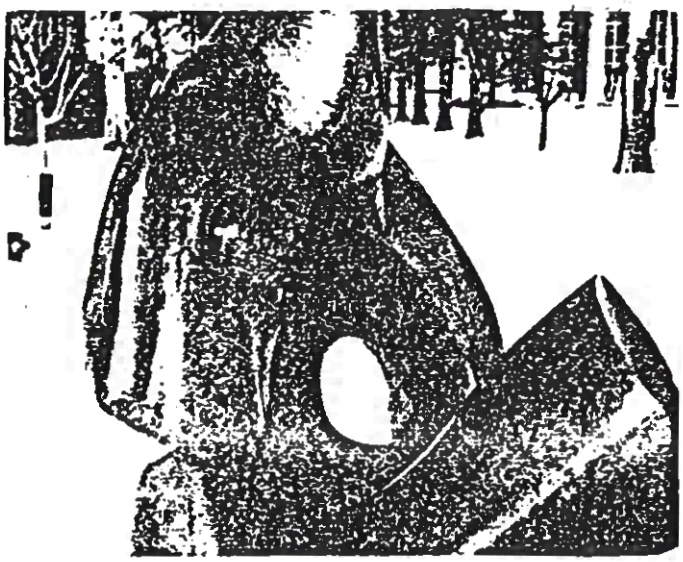
The next step is to locate changes in the differently blurred representations of the image. Again, in Section 4 we encountered Roberts cross operator which detected gradients in the image by calculating simple differences in orthogonal directions (i.e. the first derivative). However, for reasons of computational efficiency, Marr favours the use of the second derivative which is a measure of rate of change. Previously, we saw that the first derivative of a step function like change in intensity was a positive peak. Taking the second derivative instead produces a pair of peaks, one of which is positive going and the other negative going. The transition from one to the other is known as the zero-crossing:



Marr favours the use of an operator called the Laplacian operator to obtain measurements of the second derivative since the values that it produces are orientation independent. Thus the Laplacian can be applied once only to each of the arrays yielded by the use of the Gaussian filters to produce a new set of arrays containing values of the Laplacian. If the filter is a wide one, the Laplacian values will represent large scale changes in intensity in the image, while the output of a narrow one will also represent small scale changes. To detect gradients in the image, the next step is to locate the zero crossings in each representation.

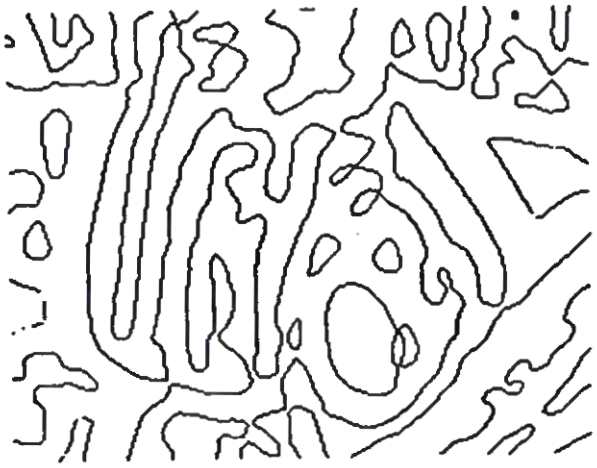
This is illustrated below, which shows the effects of using filters of three different widths.

a.



b.

c.



d.

Clearly, not all zero crossings correspond to positions of edges and

surfaces of objects in the image. So how is the edge information extracted? Marr argues that the edges of natural surfaces will be represented at various scales. This means that they will give rise to zero crossings in the output from measurements made at a range of scales. This translates to a procedure for detecting an edge segment, by looking for the presence of zero crossings in a set of independent measurements over a contiguous range of sizes of the receptive field. So, if zero crossings are found in two or more contiguous representations, and if their position and orientation is the same in each, this set of zero crossings is taken as sufficient evidence for the presence of an edge segment. By means of somewhat similar, but more complex procedure, the presence of bar segments can also be detected. An example of the use of these procedures is given below:

a.
b.
c.
d.
e.

(a) and (b) show the zero crossings obtained from the image shown above, using masks with standard deviations of 9 and 18 pixels. Because each zero crossing in (b) has a corresponding element in (a), then (b) can be taken as representing the precise location of edges in the combined description. (c), (d) and (e) show symbolic representations of the descriptions attached to the edge segments, with (c) representing blobs (i.e. closed loops of edge segments), (d) local orientations and (e) the bars. These diagrams show only the spatial information contained in the descriptors. Typical examples of the full descriptors are:

```
(BLOB (POSITION 146 21)
   (ORIENTATION 105)
   (CONTRAST 76)
   (LENGTH 16)
   (WIDTH 6))

(EDGE (POSITION 104 23)
   (ORIENTATION 120)
   (CONTRAST -25)
   (LENGTH 25)
   (WIDTH 4)

(BAR (POSITION 118 134)
   (ORIENTATION 120)
   (CONTRAST -25)
   (LENGTH 25)
   (WIDTH 4))
```

These descriptors are marked in the figure by arrows.

The set of descriptors derived from the image is stored in a database, called the raw primal sketch. But before we look at procedures for interpreting its contents, we want to understand the implications of Marr's theory for the neurophysiological evidence.

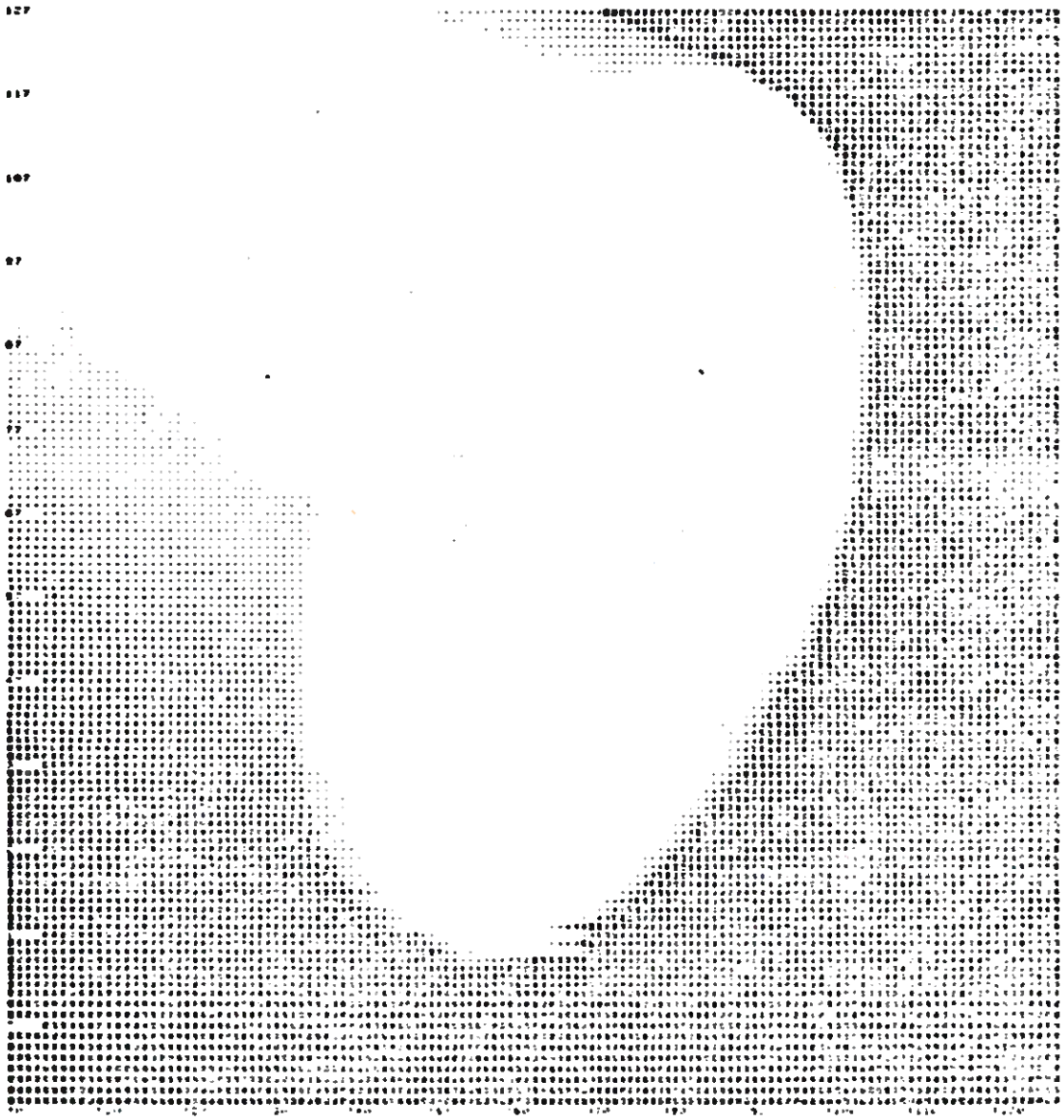
According to Marr, cells with concentric fields do not signal the presence of an edge. Instead, their function is to make measurements on the pattern of light and dark in their fields, as a basis for locating zero crossings. In practice, finding a zero crossing would involve locating activity in adjacent on and off centre cells. Finding a zero crossing segment would involve locating a set of adjacent pairs of active cells. But Marr argues that this is exactly what some simple cortical cells do.

Now, we can understand why cortical cells respond to different spatial frequencies. These are a direct result of the process of smoothing the image with a Gaussian function, using different standard deviations.

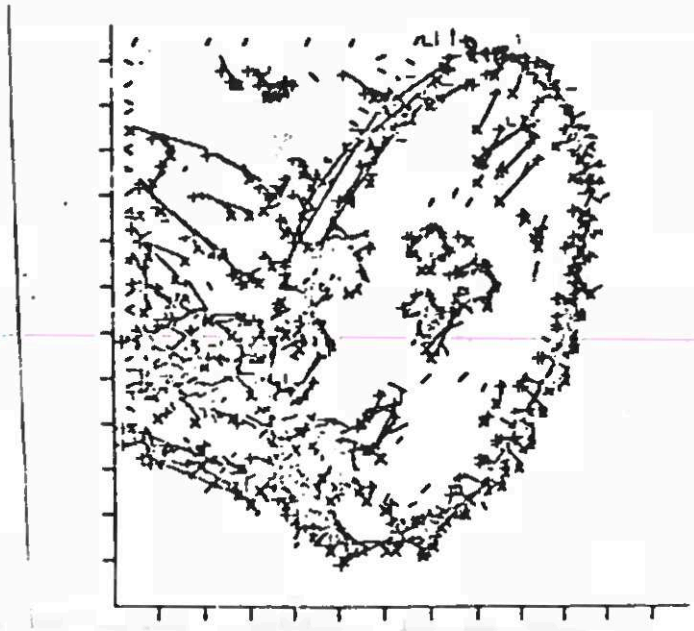
Finally, how cortical cells combine to produce the raw primal sketch is not known at present.

As stated above, the descriptors derived from an image are stored in

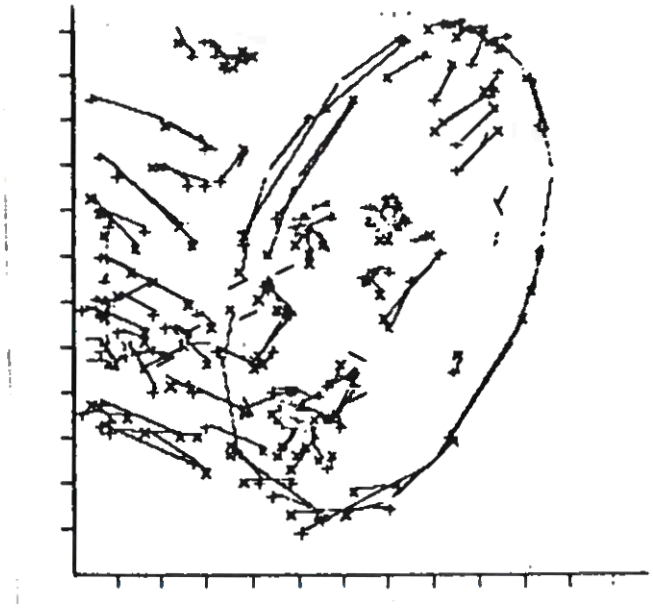
the data structure, called the primal sketch. Within it, for example, a straight line would be represented as a termination, then several segments having the same orientations, then another termination. The task is to extract these kinds of features. Unfortunately, the task is difficult due to the large number of items within the raw sketch. For example, consider the metal rod shown below:



The contents of its raw primal sketch are shown below:



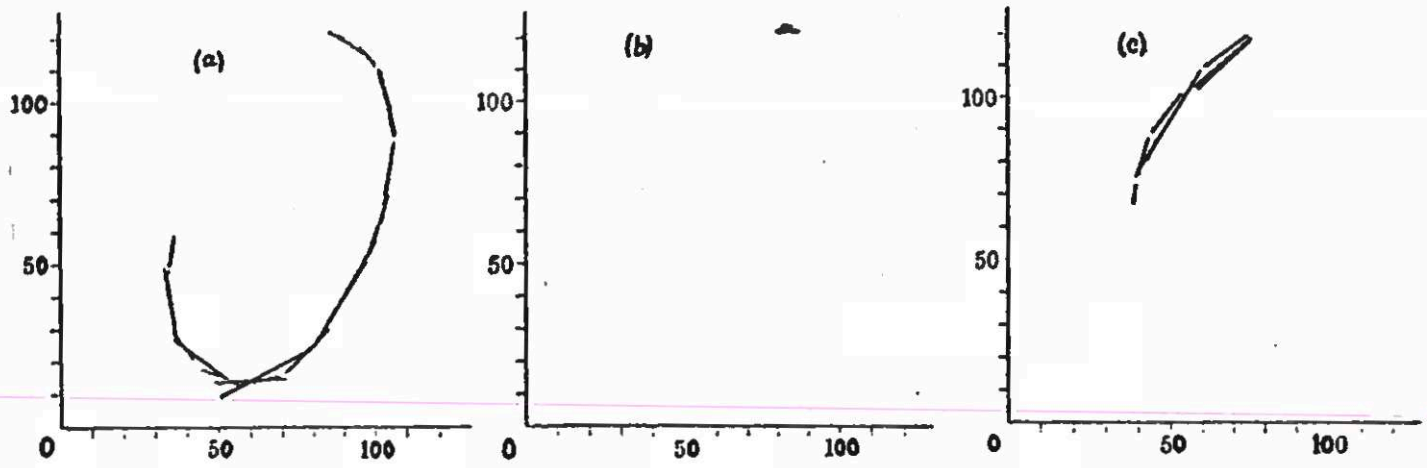
Since the rod introduces a considerable amount of extrinsic noise, the data-base is large. However, much of the noise can be filtered out by applying a very simple rule, namely that a short segment's assertion is eliminated from the data-base if (a) it crosses a longer segment, and (b) its contrast is less than that of the longer one. The effect of applying this rule to the data-base of assertions is represented graphically below:



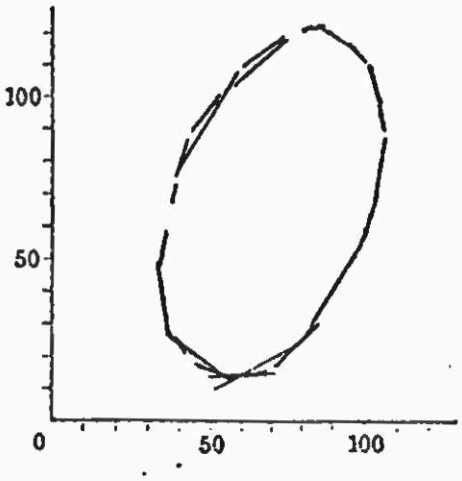
We know where to look to find the edge between the side of the rod and its end. It stands out quite clearly, making the task appear to be quite easy. How does the program handle it? The answer is that it applies a grouping technique called curvilinear aggregation which is a three-stage process. The first stage of grouping combines two elements in the primal sketch only if they match in almost all respects, are very close to one another, and if there are no other candidates. The information that is used by the process to determine whether or not two items should be grouped include orientation, contrast, type (edge, bar , etc.), fuzziness, distance between nearest parts of the two items and orientation of items relative to orientation of a line joining their nearest parts.

The second stage of grouping makes use of the extra information given by the first. For example, some segments are now quite long (more than 20 image elements). Two such elements may be combined, even if the value of some of their parameters differ, provided there are no other near candidates. Also, new orientation information may be available as a result of the first grouping, to be used as an important parameter in second stage grouping.

When the first two stages of curvilinear aggregation are applied to the primal sketch of the rod they produce the larger elements shown below:



The third stage of grouping is rejecting unlikely possibilities. A node is set up for each of the ends of the segments delivered by the preceding processes. With each note is associated a list of nodes that could possibly match it. Each possible match is evaluated independently against the criteria, and possibilities that are graded relatively poorly on several counts, and well on none, are rejected. Nodes at which ambiguities exist are marked, this information being sent to the next level of processing. The results of applying this third stage to the primal sketch of the rod are shown below where the elliptical form of the contour is shown.



Up to this point in the analysis, the system has not used any description of the form's overall shape. The fact that it has been able to recover the contour information is evidence in favour of Marr's claim that a richer description will enable a system to extract form information from 2-D pictures of objects which have not been specially treated to reduce extrinsic noise. This is a very considerable achievement if validated by results from a wide range of scenes of different kinds.

II. INTERMEDIATE LEVEL ANALYSIS

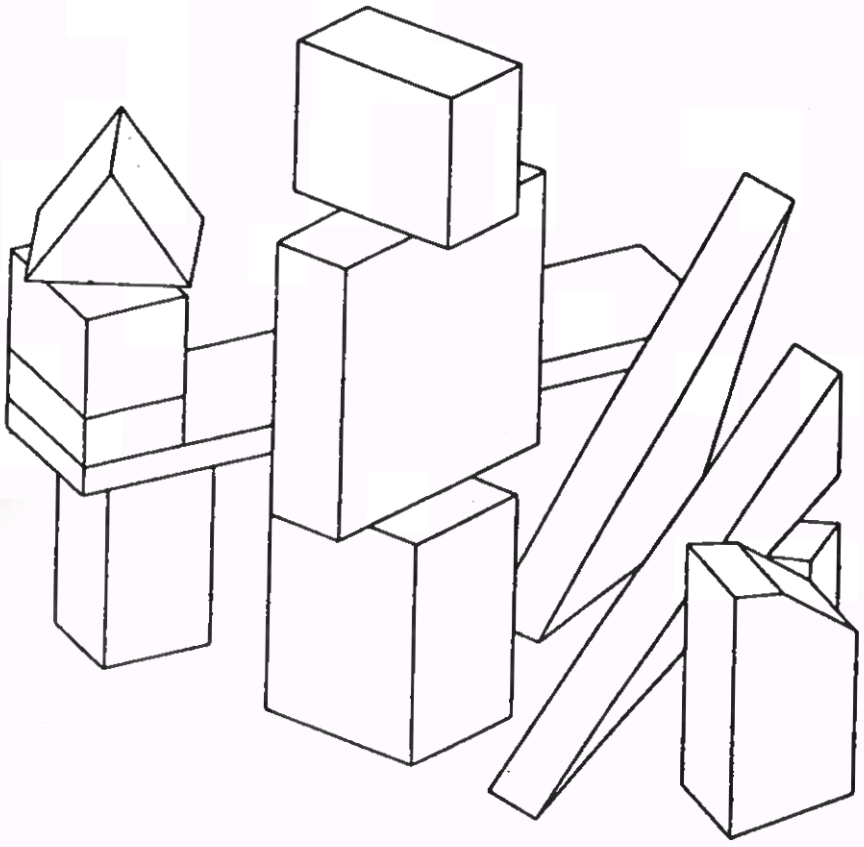
THE SEGMENTATION PROBLEM

The next task is to find what structures are present within the mass of feature information provided by all hypercolumns together, so that these structures can be recognized by comparing them with descriptions stored in memory.

The problem of finding structures within a feature description is the problem of deciding which features belong together and which do not. This is the segmentation problem.

For many years, psychologists have been interested in the problem of how the human visual system groups the incoming visual data to represent the objects in the visual field. A group of German psychologists, known as the Gestalt psychologists, were particularly interested in the problem, and they enumerated a number of laws of organization. We have already encountered three of their laws, namely similarity, proximity and continuity, when discussing Marr's methods for extracting global features from the primal sketch representation, i.e. for generating descriptions of objects in a scene.

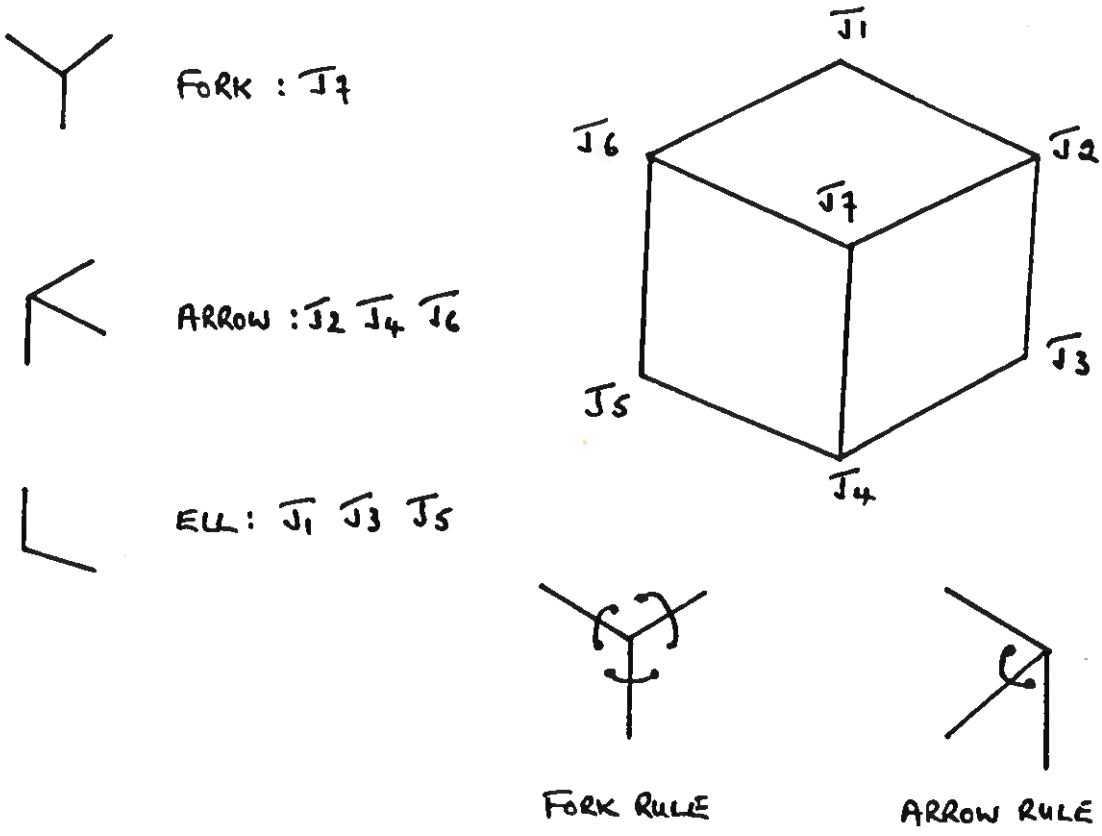
Now suppose that the scene contains a jumble of blocks of different sizes and shapes, as shown below.



Assuming perfect input data, the artificial system would extract a line representation of the edges of the objects, in terms of a list of (x,y) end points and associated line orientations:

e.g. (2,3) (2,11) 270°
- - - - - etc

Of course we can tell how the regions combine to form 3-D objects, and how many objects there are in the pile. The question is how does the visual system do it? We will begin by looking at some of the characteristics of 2-D drawings of planar solids. Taking the cube shown below as example, notice that its three faces meet at a trihedral vertex.

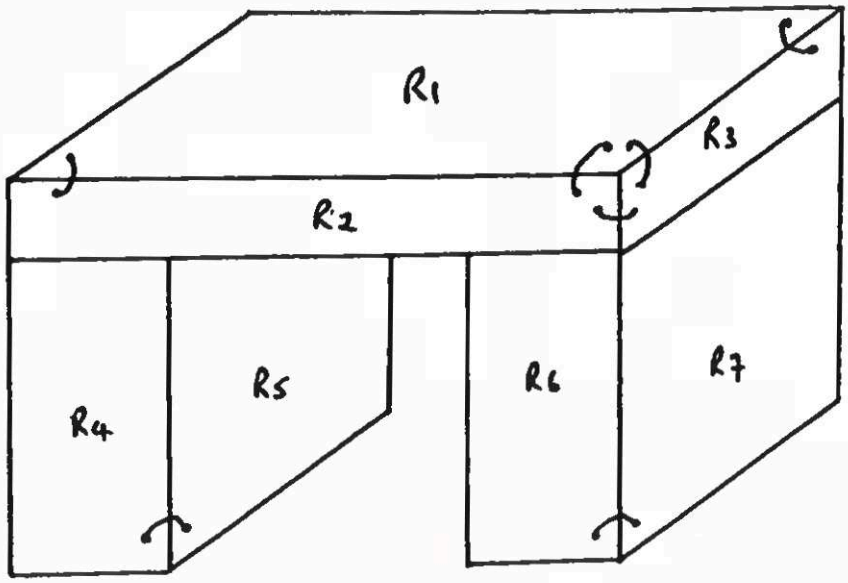


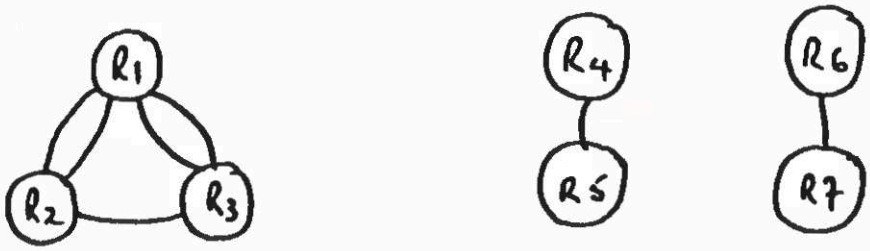
In a 2-D drawing of a cube, the three edges forming such a vertex are represented by the junction of lines, forming either a FORK junction: J7 or an ARROW junction: J2 J4 J6 or an ELL junction : J1 J3 J5, as shown above. Notice that the number of visible faces at each vertex determines what the

junction will look like:

3 visible faces produces a FORK junction in the picture.
2 visible faces produces an ARROW junction in the picture.
1 visible face produces an ELL junction in the picture.

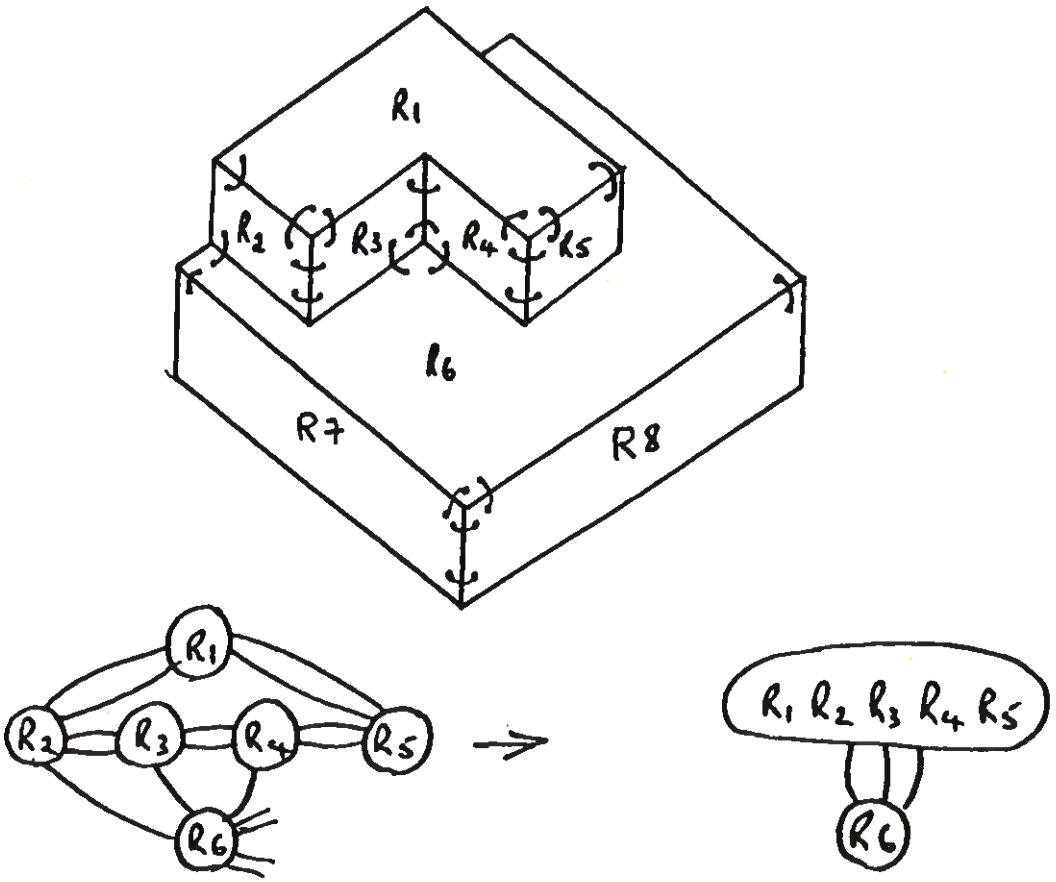
So far, we have been going from 3-D to 2-D. But notice that if we are given a 2-D representation of a collection of planar solids, we can decide which regions belong to which solids using rules. For example, the FORK rule links together all three regions surrounding a FORK junction, and an ARROW rule links together two of the regions contributing to the junction. For example, to segment the 2-D line drawing of the arch shown below into its component parts, links are planted between regions wherever an ARROW or FORK occurs. On the basis of these links, the regions can be collected together into three groups, namely (R1 R2 R3) (R4 R5) (R6 R7), where each group represents one of the bodies making up the arch.





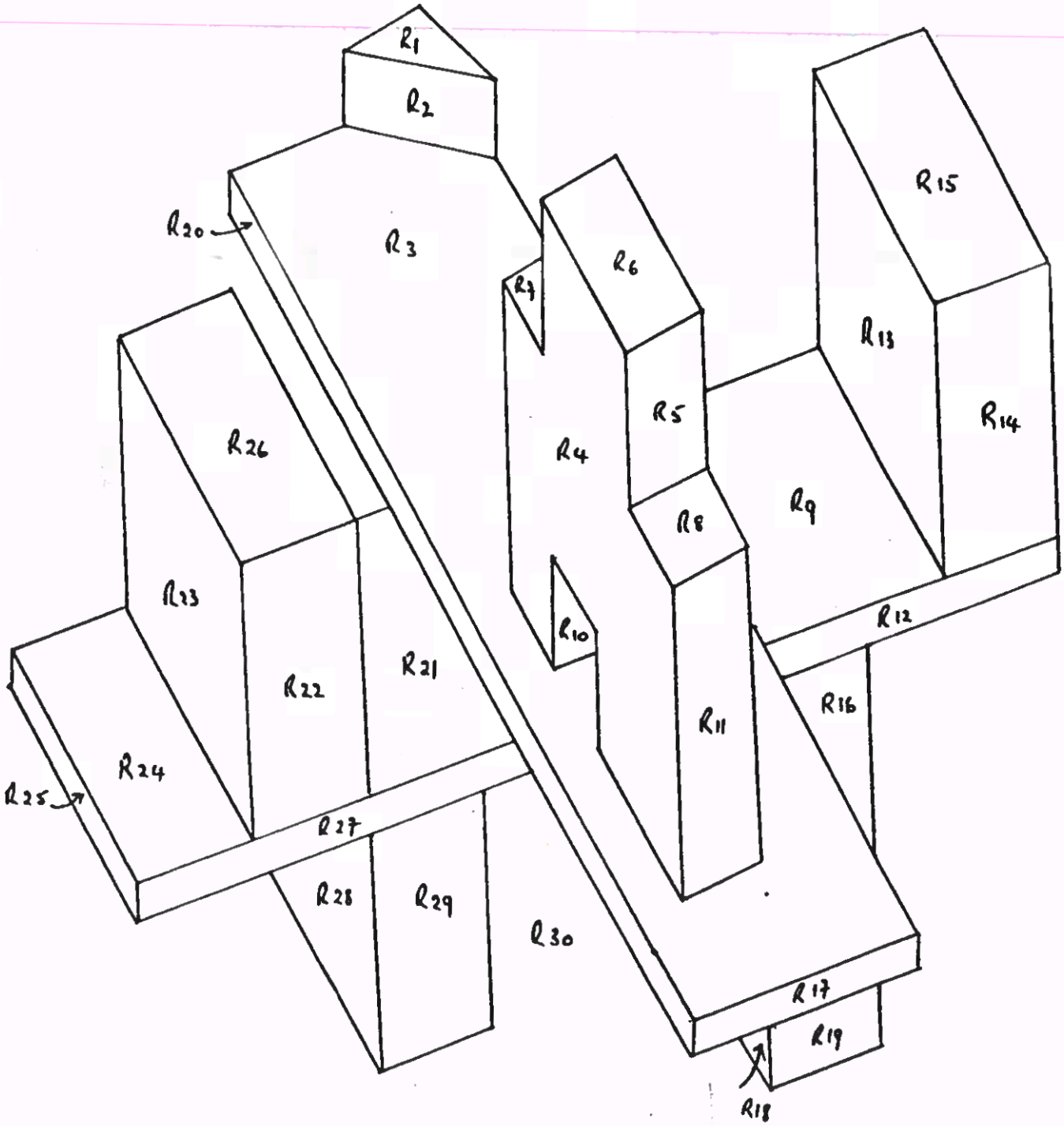
Notice that the other type of junction most prevalent in the arch scene is the T-junction. As you can see, at a single T-junction no rule can be made that the three separate regions are to be joined as parts of the same body. However, when T-junctions can be grouped together in pairs, this can provide powerful evidence of the interposition of one object in front of another one. For example, if you place your thumb on the edge of the table in front of you, you will occlude its edge and create two T-junctions, where the edge of the table meets and leaves the edge of your thumb. In this case, the stems of the T-junctions are in line i.e. they are collinear, so the regions on either side of the stems of the T's can be linked.

But the following example suggests that the task is more complicated than it has appeared so far. Instead of segmenting Regions R1 to R8 into two bodies (R1,R2,R3,R4,R5) and (R6,R7,R8), all regions are connected together by multiple links.



It was this task that Guzman tackled in implementing a program called SEE. We will examine SEE in some detail since it was the first of a series of programs, each of which built on the ideas and experiences with the previous one, gradually reducing the need for ad hoc rules by providing a better theoretical justification of the underlying processes.

In SEE, Guzman assumed as starting point the existence of a perfect line drawing of a polyhedral scene. A typical example is the scene called BRIDGE, shown below. This is input to the program in the form of unordered lists of object regions, background regions and vertices. Notice that the program does not have to separate objects from background: this information is provided by Guzman.

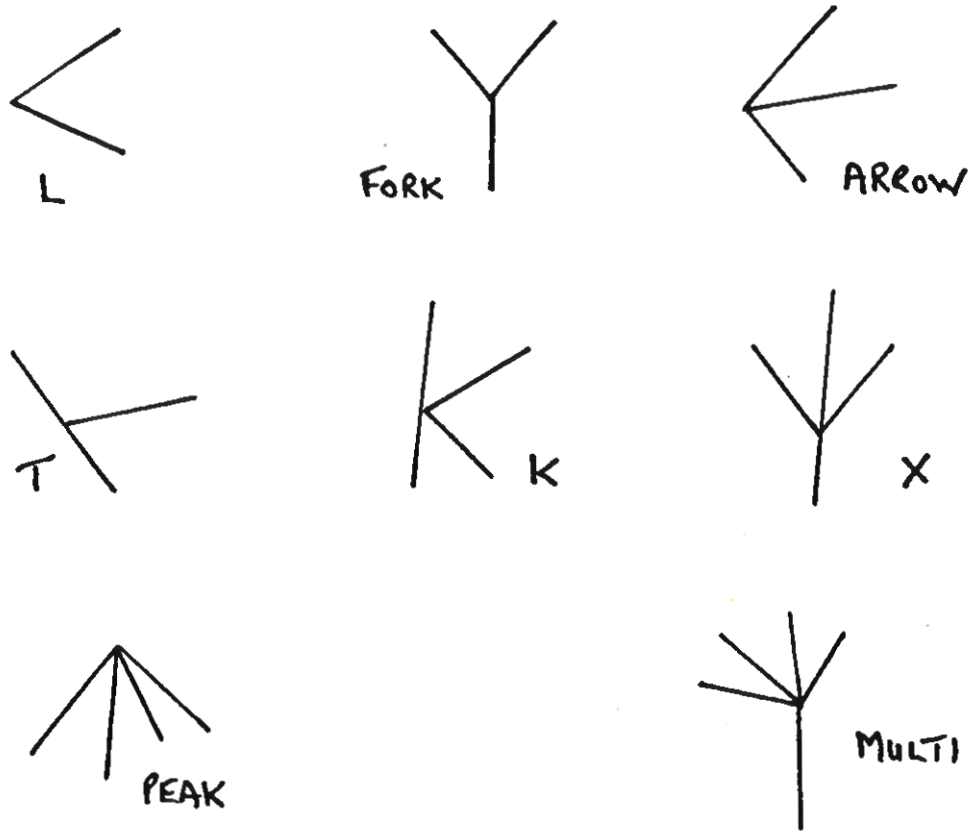


To parse the scene into bodies, SEE follows a two part strategy. First, it collects evidence for linking regions. Then, second, it evaluates this evidence and groups regions to form objects.

We will begin by considering the first part strategy, namely collecting evidence. It was Guzman who noticed that, as we discussed above, the shape of a junction is a pretty reliable indicator of its three-dimensional significance. In practice, Guzman classified junctions into four basic types:

1. Vertices where two lines meet, e.g. L
2. Vertices where three lines meet, e.g. ARROW, FORK, T
3. Vertices where four lines meet, e.g. K, X
4. Other vertices, e.g. PEAK, MULTI

Examples are shown below:



With each type of vertex there is an associated set of links which constitute the evidence for combining adjacent regions in the scene. These links are of two types, namely strong and weak links. The strong links associated with each vertex are as follows:

1. Ls, Ks, MULTIs and single Ts have no links.

2. FORK. Links are planted between the three regions, meeting at a vertex of the FORK type, except:

   (a) if one region is the BACKGROUND no links are placed between any regions surrounding it.

   (b) if one of the lines is connected to an L, or to the barb of an arrow, or forms the bar of a T, the regions on either side of that line are not linked.

3. ARROW. Links are placed between the regions on either side of its shaft, except

   if the shaft of the ARROW is connected to a background FORK, or to the stem of a background T, the regions on either side of each of the barbs are linked.

4. X. Two cases are distinguished

   (a) If the X is formed by the intersection of two lines, no links are planted.

   (b) If the X is formed by four lines, two of which are collinear, the regions on either side of the collinear lines are linked.

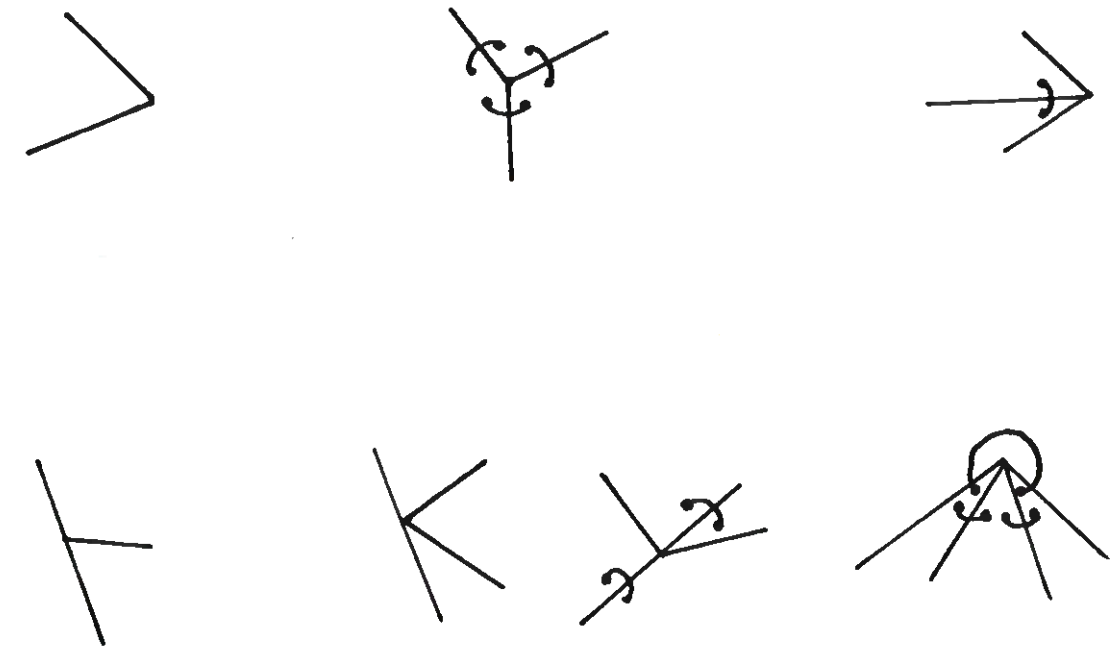
5. PEAK. All regions, except the one containing the obtuse angle, are linked to each other.

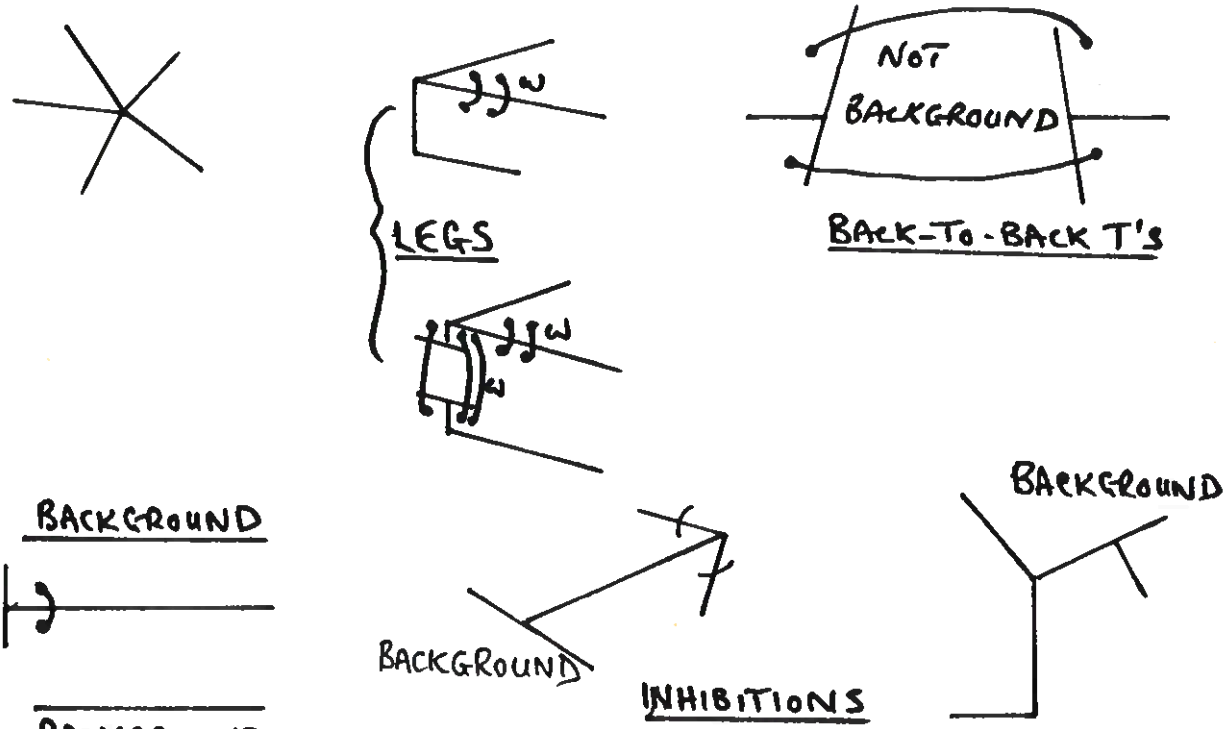
6. T pairs. Facing pairs of Ts with collinear stems are linked, provided the area between the bars is not BACKGROUND.

7. 3-parallel T. The regions on either side of the stem of the T are linked in the case of a 3-parallel T.

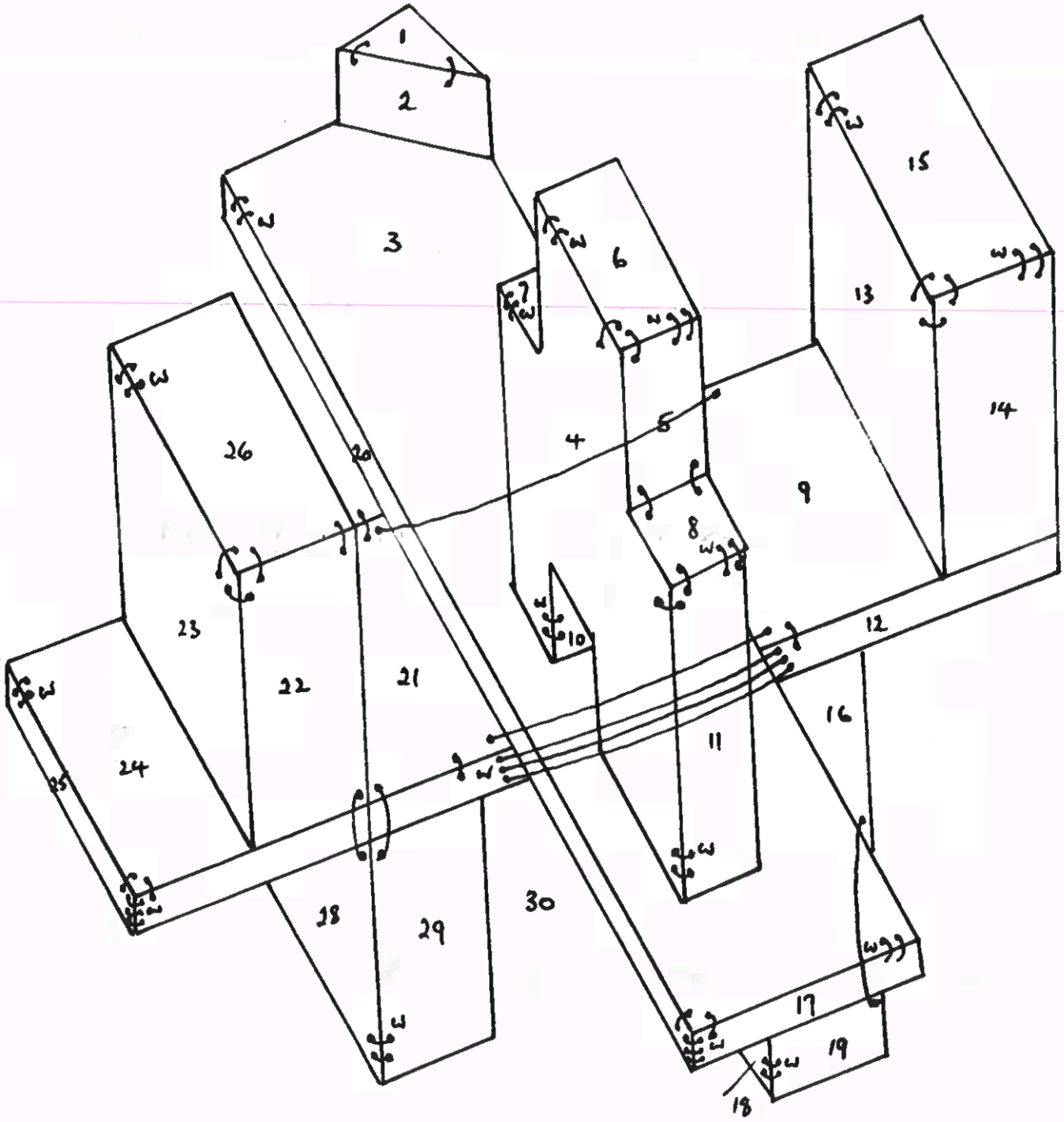
Weak links, planted in addition to strong links, are associated with the type of vertex called LEG. LEG is an ARROW where one of the barbs of the ARROW is connected to an L which has one line parallel to the shaft of the ARROW (if necessary through a chain of matched Ts).

Examples of the links associated with these junction types are given on the next page.

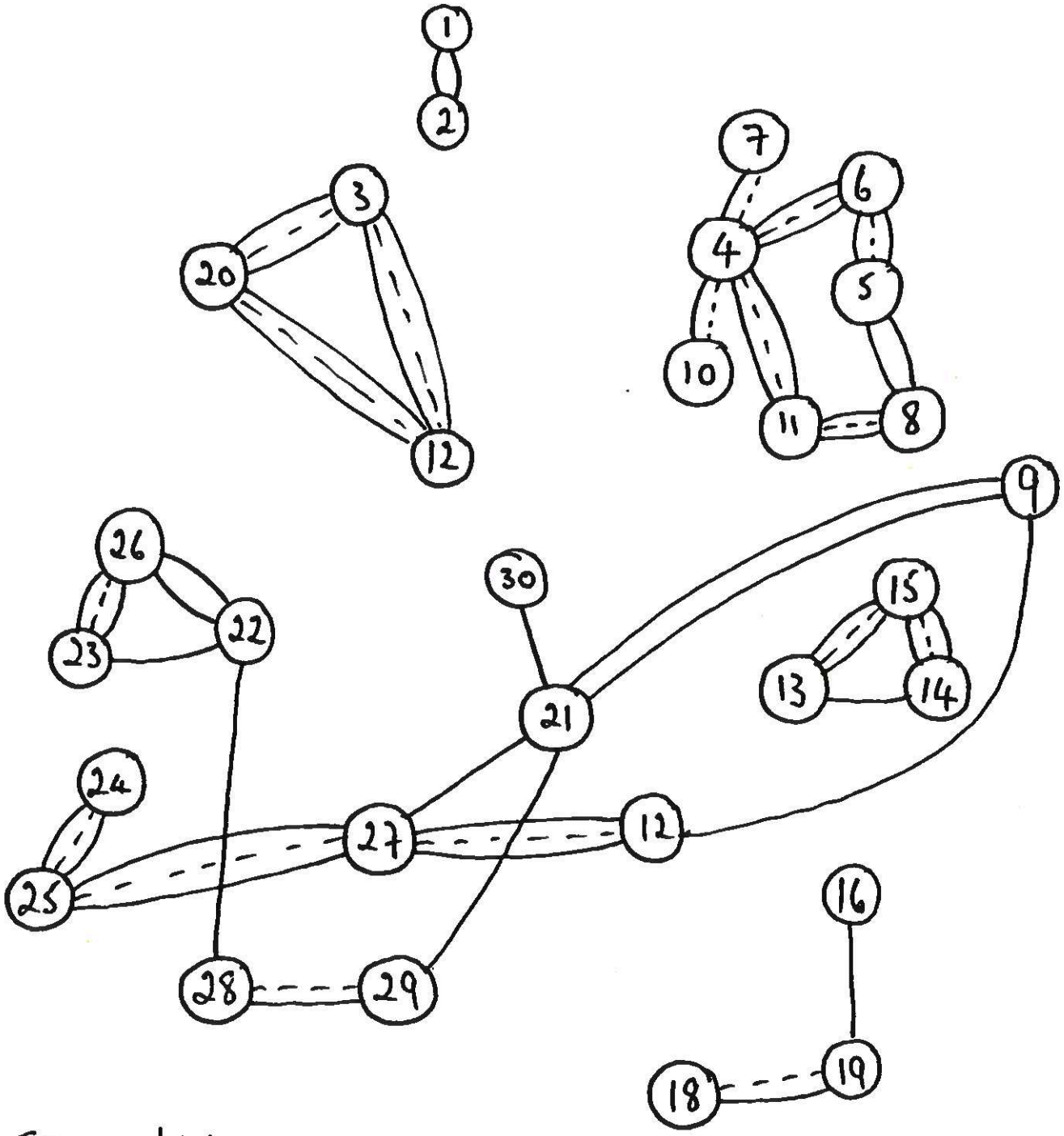



LEGS
NOT
BACKGROUND
BACK-TO-BACK T's
BACKGROUND
BACKGROUND
3-PARALLEL T
ON BACKGROUND.
BACKGROUND
INHIBITIONS
BACKGROUND

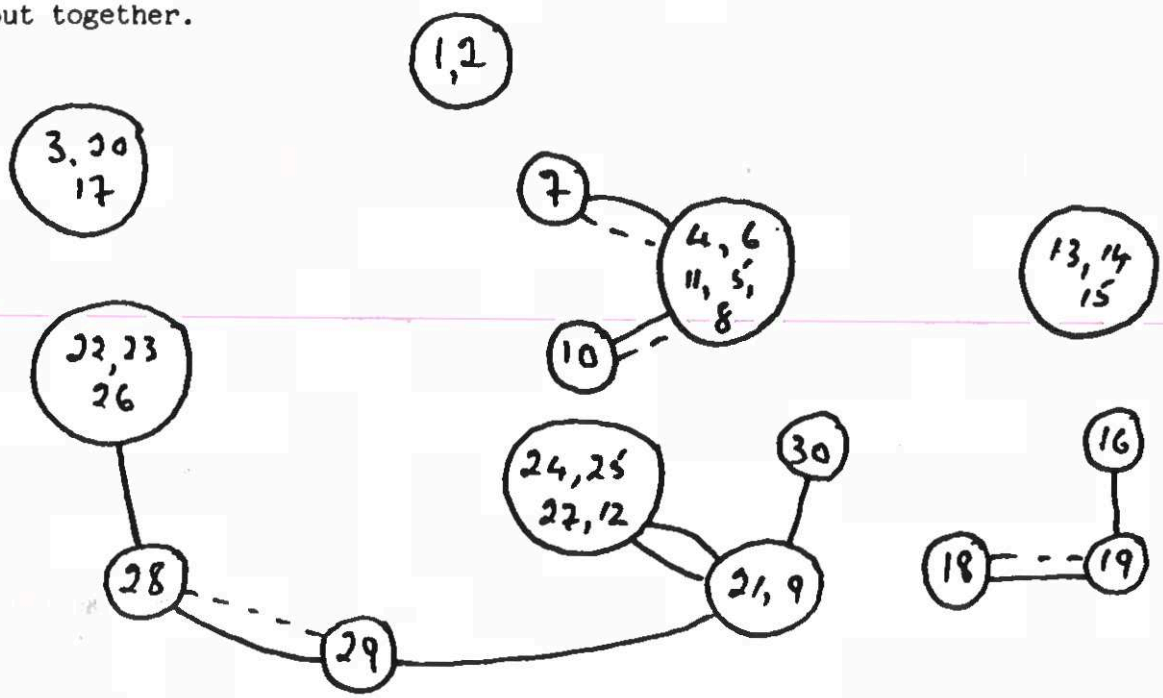
The strong and weak links associated with the scene BRIDGE are shown below:



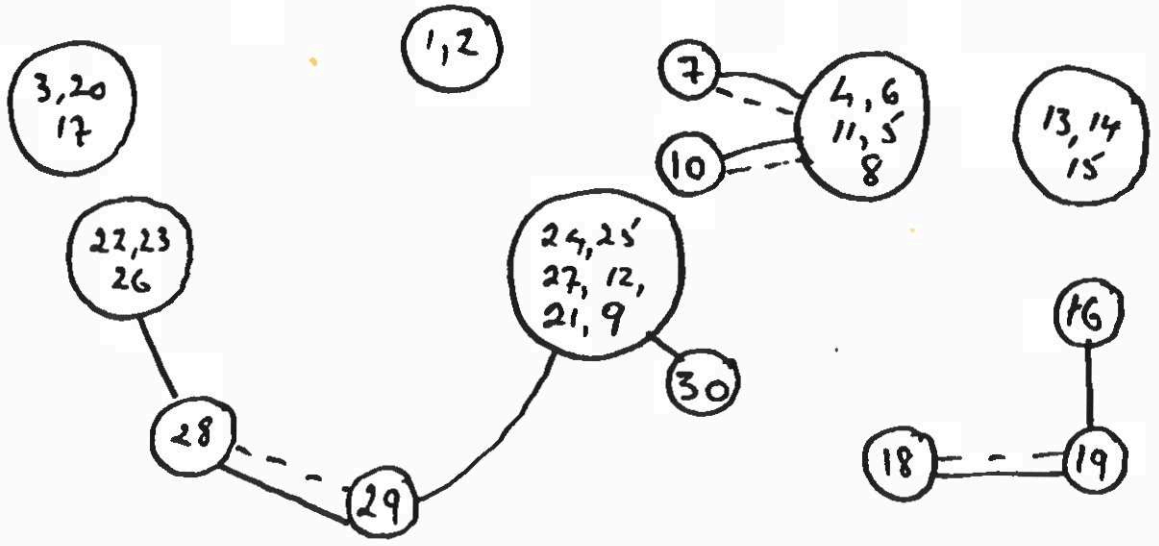
The second step is to combine and group the link evidence to partition the scene into its constituent bodies. The evidence for the scene BRIDGE is shown below, in which the regions are depicted by circles. Strong links are represented by solid arcs, weak links by dotted arcs. All the links to the background (:30) have been deleted since the background cannot be part of any body.



Now the program attempts to form nuclei, where a nucleus is either a region or a set of nuclei formed by the following rule: if two nuclei are connected by two or more links, they are combined to form a larger nucleus. For example, as shown next, regions :24:25:27:12 and regions :21 and :9 are put together.

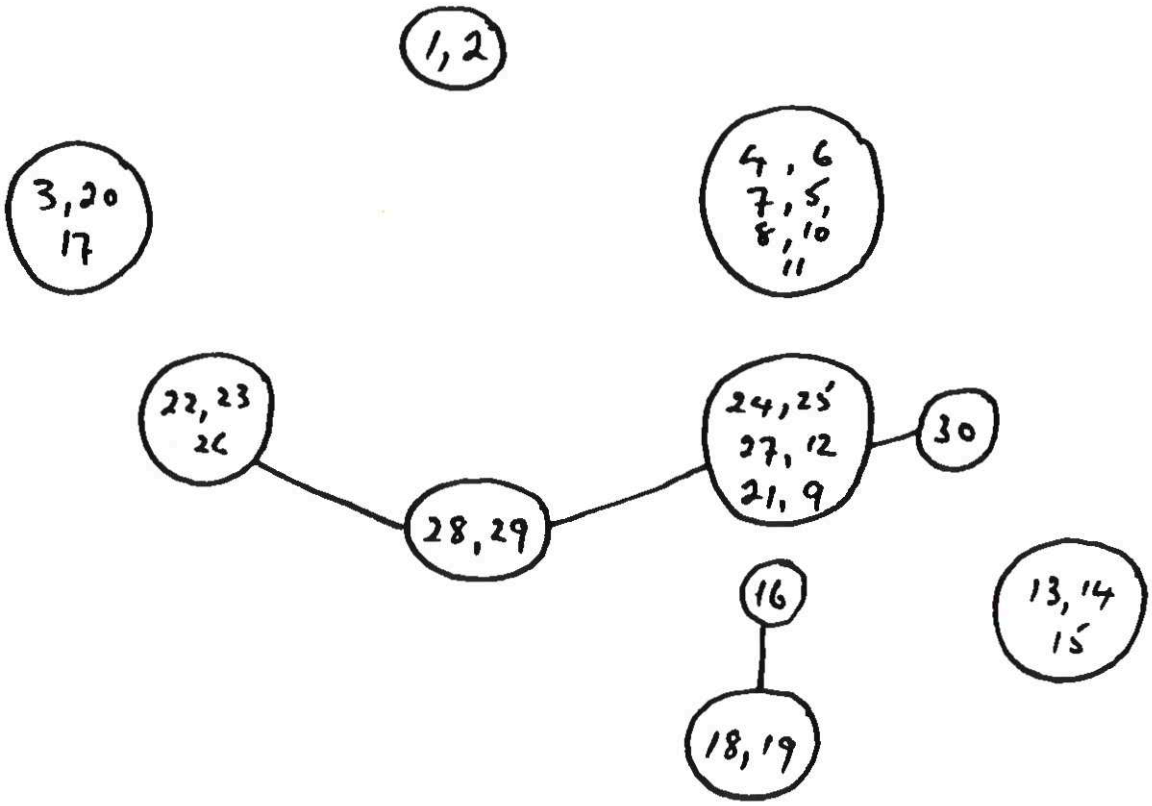


As a consequence, nucleus :24:25:27:12 has two links with nucleus :21:9, so they are combined in turn to form a new nucleus :24:25:27:12:21:9, as shown below:



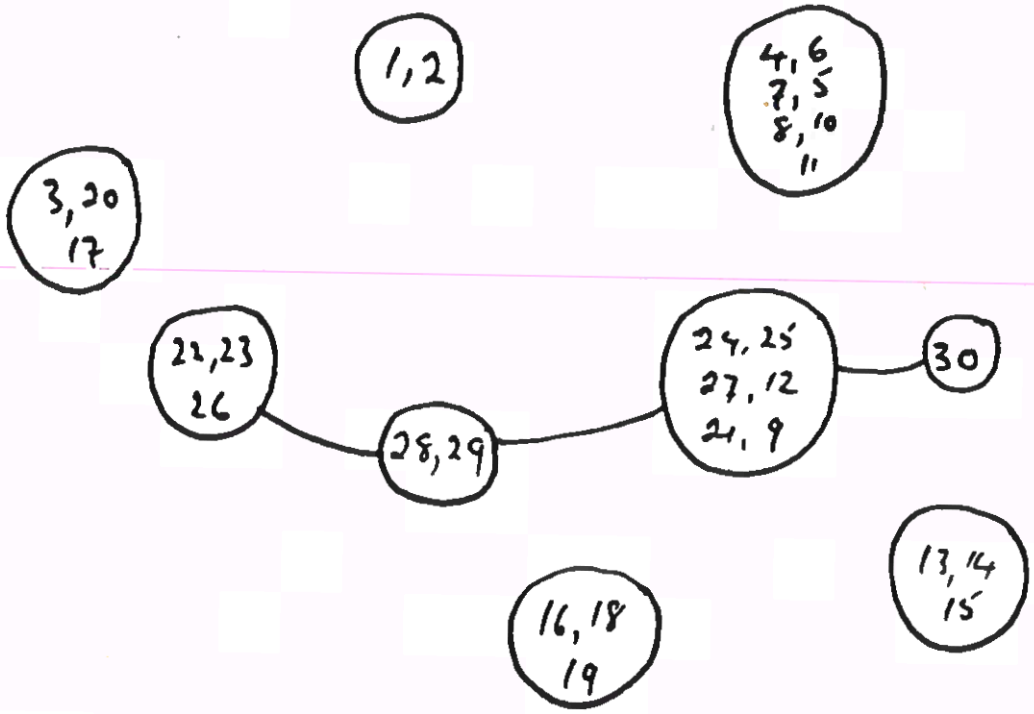
So the nuclei are allowed to grow and merge until no new nuclei can be formed. When this is the case, the scene has been partitioned into several "maximal" nuclei: between any two of these, there are zero or, at most, one link.

The program has still to consider the effect of weak links. The rule is that if a strong link joining two maximal nuclei is reinforced by a weak link, these nuclei are merged, as shown next.



For example, in scene BRIDGE, the following weak links exist: :13 to :15, :14 to :15, :14 to :15, :3 to :17, :7 to :4, :8 to :11, :10 to :4, :5 to :6, :28 to :29, :18 to :19, :25 to :27, :22 to :26, :23 to :26.

Notice that nucleus :16 is linked to nucleus :18/:19 by a single strong link. This invokes another rule to the effect that a strong link joining a nucleus and another nucleus composed by a single region is sufficient evidence for the nuclei in question to be merged if there is no other link emanating from the single region. This yields the final parsing shown below:



In summary:

i. Form nuclei from regions connected by two or more strong links.

ii. Amalgamate nuclei joined by two or more links until no new nuclei can be formed.

iii. Amalgamate nuclei joined by one strong and one weak link.

iv. Amalgamate a nucleus jointed to a single-region nucleus by a strong link (except when the single region is BACKGROUND). Ignoring the single links between nuclei which remain after parsing, the program returns the results:

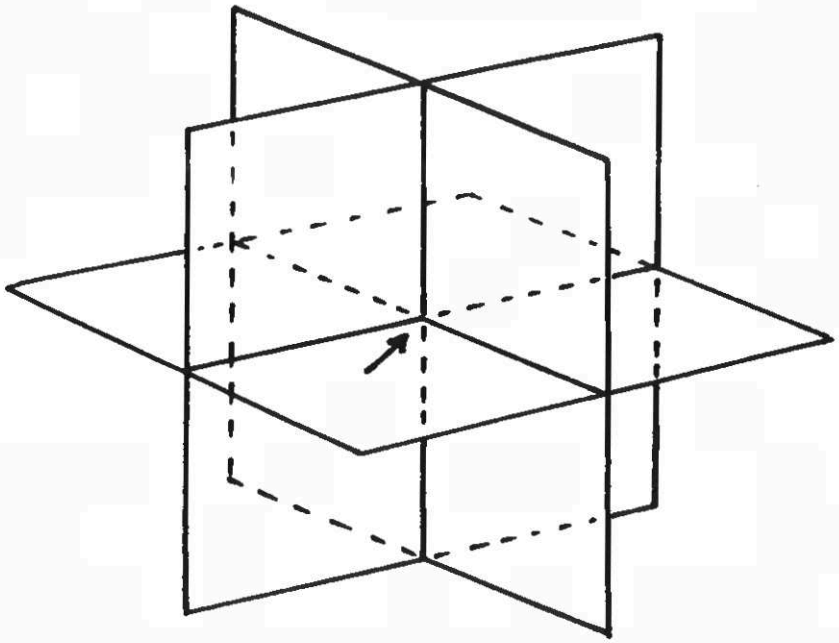
```
(BODY1.  IS  :24  :9   :21  :27  :12  :25)
(BODY2.  IS  :22  :26  :23)
(BODY3.  IS  :17  :3   :20)
(BODY4.  IS  :1   :2)
(BODY5.  IS  :14  :15  :13)
(BODY6.  IS  :19  :18  :16)
(BODY7.  IS  :29  :28)
(BODY8.  IS  :8   :11  :5   :6  :4  :10  :7)
```

How good is SEE? Since it requires two pieces of strong evidence to join two nuclei, it is conservative, i.e. it will almost never join two regions that belong to different bodies. Its errors are almost always of the same type: regions that should be joined are left separate. This suggests that more heuristics should be added to provide additional linking evidence. The problem is that adding a heuristic can cause repercussions: it may solve the difficult case but in turn cause other difficulties. Rather that continue to derive rules in an ad hoc way, it would be preferable to derive them from an explicit 2D/3D representational theory which takes into account the overall geometry of polyhedral bodies. This is what we will consider next.

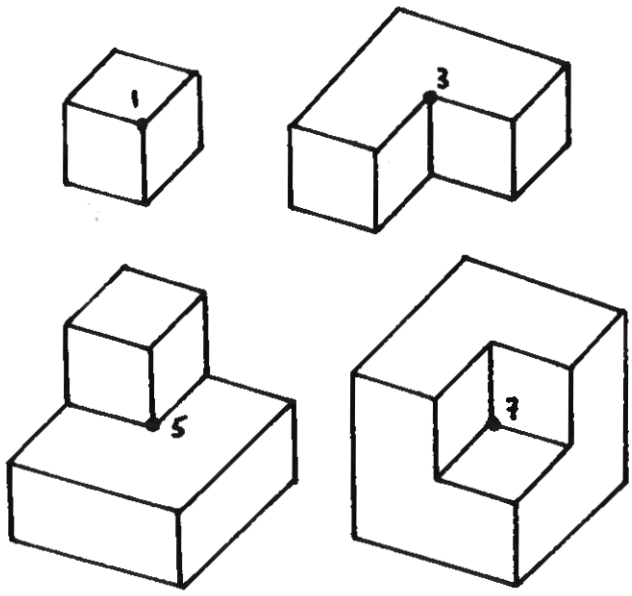
EXPLOITING PHYSICAL CONSTRAINTS

When we discussed Guzman's program, we found that its rules for linking regions depended on the shape of the local junction. In contrast, we turn now to consider later work by Huffman, Clowes and Waltz who realised that by devising rules for describing and linking junctions, not only could they obtain a segmentation of the scene into bodies, but they could also derive information about the 3-D shape of the bodies.

As we have already noted, SEE made most use of trihedral vertices - the so-called ARROW and FORK junctions. Now, a trihedral vertex is a point of intersection of three planes which partition the surrounding space into eight octants. This is shown below:



Some of the ways in which these octants can be filled by three surfaces which meet at a vertex are shown overleaf, where the number of octants actually occupied yields a type number. For example, type 1 is like Guzman's ARROW, and type 7 is like his FORK junction.

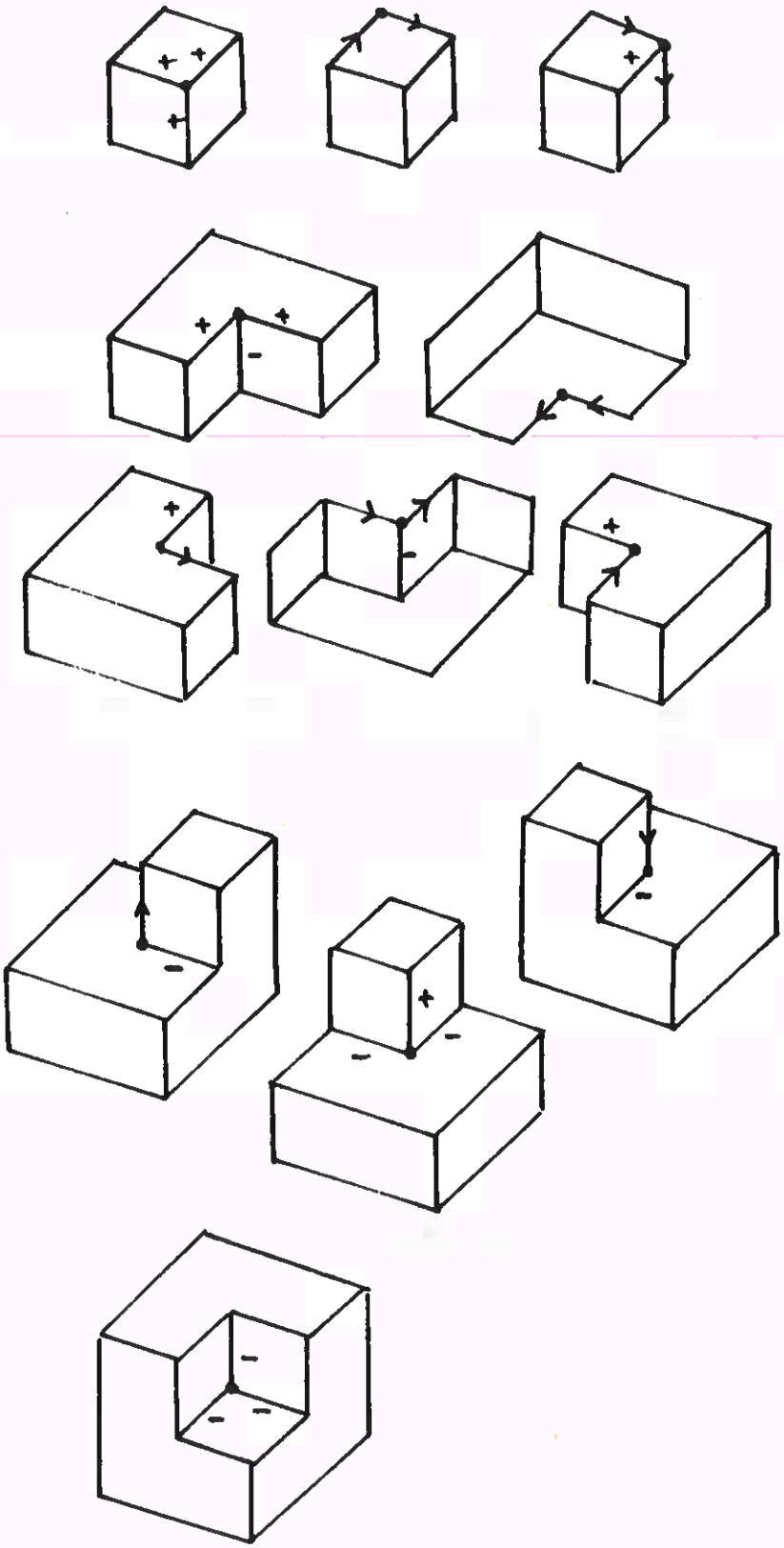


Imagine now that you can view the vertex from each unoccupied octant. The possible views of this vertex are shown opposite. Labels are associated with lines in these drawings. Let's see what these labels denote:

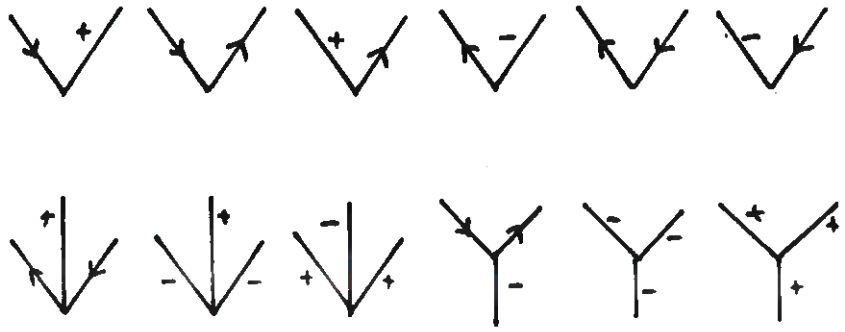
1. a '+' marks a convex edge which has both corresponding planes visible.

2. a '-' marks a concave edge which has both corresponding planes visible.

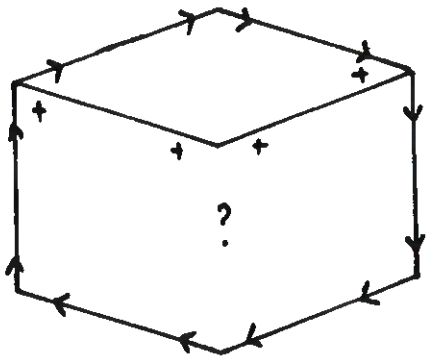
3. an '<' marks an occluding edge where one plane is hidden, the visible plane being to the right of the direction in which the arrow is pointing.



By understanding the three-dimensional nature of a scene, we are able to apply these labels to the 2-D line drawing. The important question is can a program use these labels to help it understand the three-dimensional nature of line drawings? The answer is that it can do so by labelling to the vertices in a 2-D drawing in accordance with the set of labelled line configurations which we obtained by labelling the possible views of the vertex. The set of twelve possible configurations is shown below.
Notice how this approach limits the number of labellings for the different configurations. For example, given four labels, there should be 16 ways of labelling an "L" junction but there are only 6 legal labellings shown.

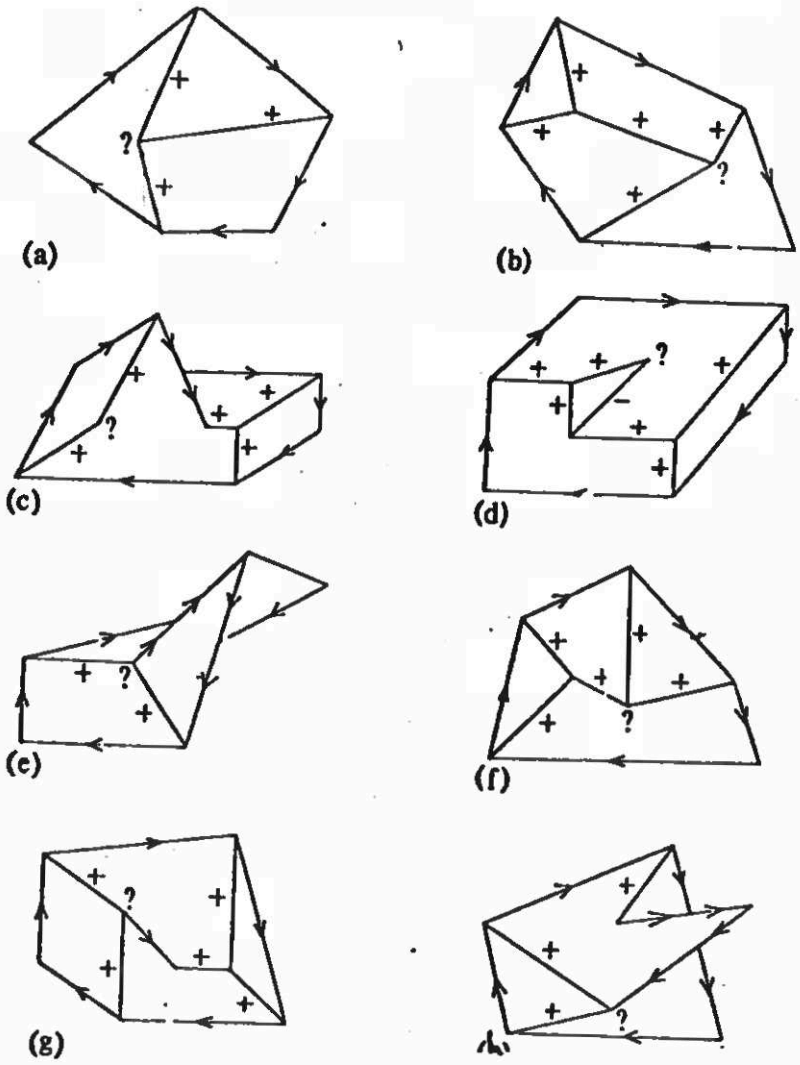


Huffman was interested in showing that the use of the labelled-line configurations (which we will refer to as corner models) would enable us to tell when certain kinds of drawings are impossible. If we look at the drawing given below, it will be rejected as a possible plane-faced object because there is no set of labels which will consistently label its 2-D line representation.



Labelling the outer contour is straightforward - the only allowable labels are arrow-type labels. If we move on now and consider the two ARROW vertices, we find that arrow labels have been assigned to the lines (barbs) on either side of the shaft in both cases. Inspection of our list of legal corner models, given above, shows that there is only one ARROW vertex with arrow labels assigned to its barbs. Selecting this forces a + label for the shaft which is entered accordingly. If we now consider the L vertex between the shafts of the two ARROWS, we find that each leg of the L has been assigned a + label. But inspection of the list of corner models indicates that this is <u>not</u> a legal corner model - there is no L configuration with + labels on each leg so we conclude that the drawing does not represent a regular plane faced object.

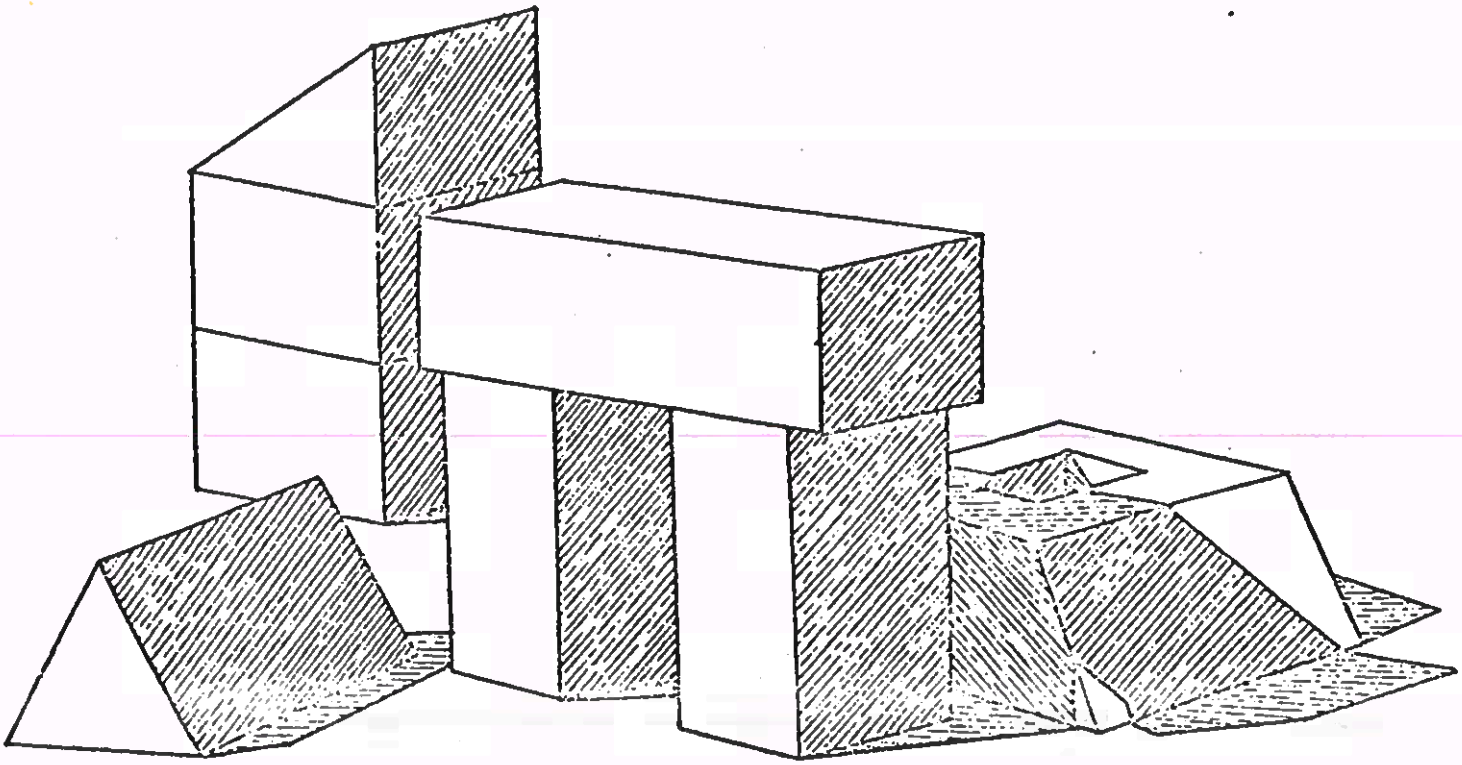
Similar considerations apply when we examine these eight objects:

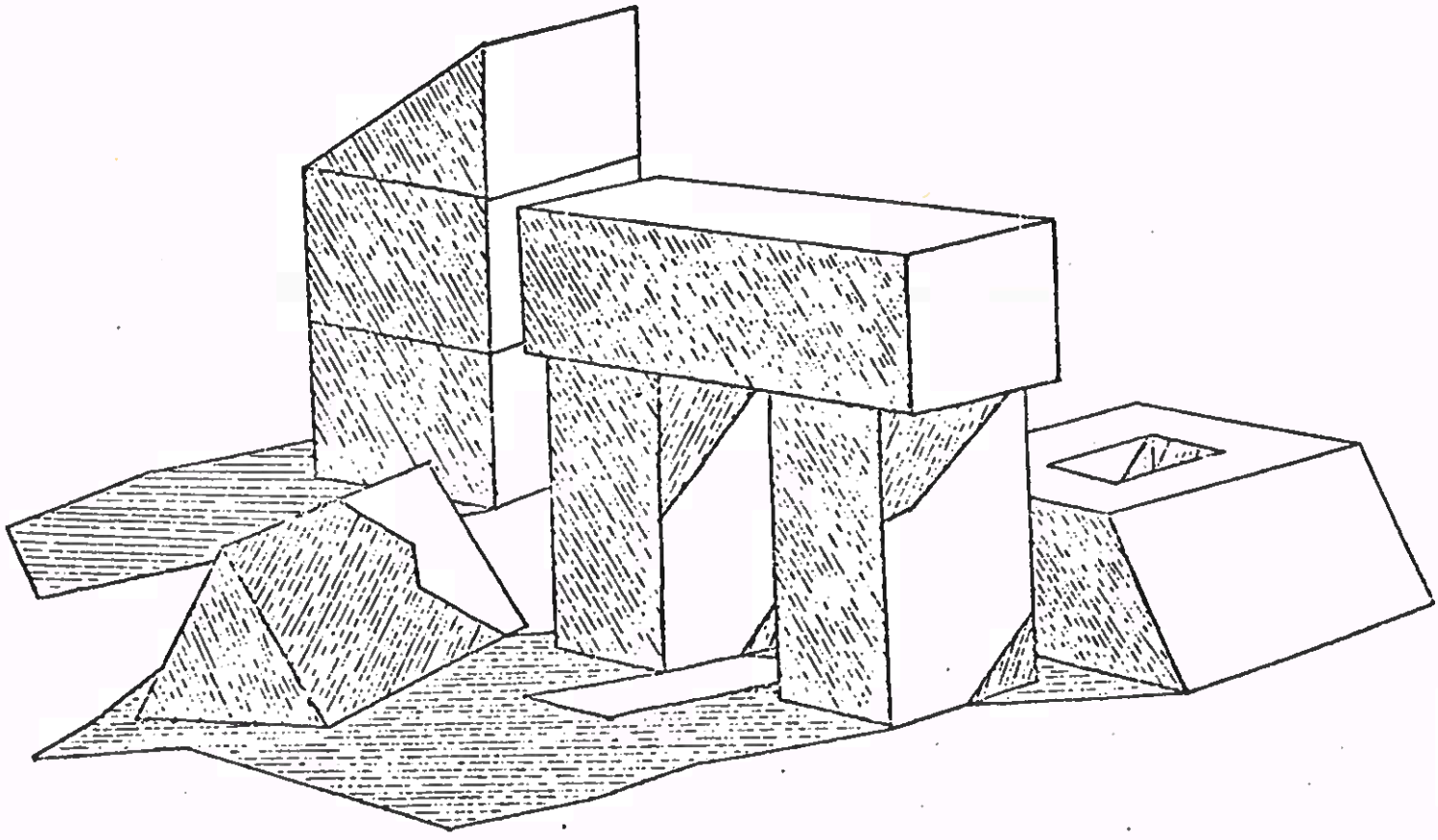


In the case of example (a), the labelling of the outer contour forces us to label the shafts of the ARROW with a + since this is the only legal corner model which can be applied. This forces + labels on all three lines forming the ARROW in the middle, and inspection of the list of corner models indicates that it is not a legal labelling. Exactly the same problem crops up in examples (b) and (f), and in the case of example (c) we see a recurrence of the labelling problem encountered in the case of the drawing of the incomplete cube, seen earlier. Example (d) is a second example of an illegal L model, whereas example (e) has an illegal FORK junction. Examples of other types of illegal ARROW labellings are shown in examples (g) and (h).

Huffman only considered single objects, using a hand-worked analysis. Clowes, working independently on the problem, devised a computer program, called "OBSCENE", to perform this kind of analysis. Since it was designed to handle scenes with multiple objects, involving consideration of additional fork and T-junctions, Clowes' program was equipped with a larger set of corner models.

Working at M.I.T., David Waltz generalized the Huffman/Clowes ideas in two fundamental ways to handle scenes like those shown overleaf.



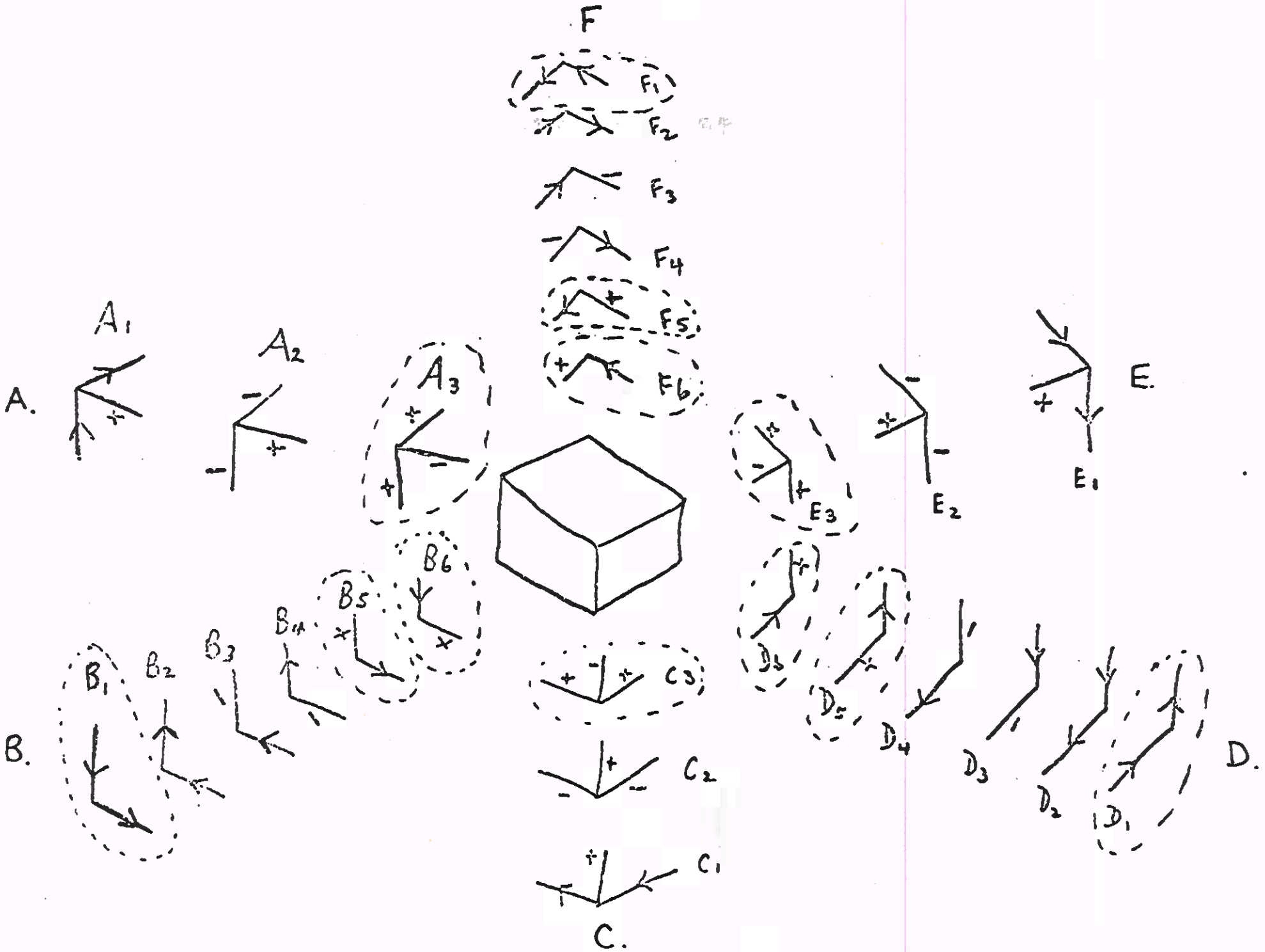


1. To capture more information about the physical situation, he expanded the Huffman/Clowes line labels to include boundary edges, convex edges, concave edges including separable concave edges, crack edges and shadow edges. After expansion the total number of legal corner models in the data-base was 717. This stands in marked contrast to the many thousands of possible corner models that could be generated, given that each edge could be labelled in 12 possible ways. In other words, the structure of the scene severely constrains the number of alternatives (i.e. restricts the search problem).

Not content with this, Waltz also extended the labels to include information about illumination. The surfaces of objects can be categorized as Illuminated, Self-Shadowed or Shadow-Projected: This means that a given edge type can have illumination information added which describes the illumination category on either side of that edge. For example, if a concave edge is Illuminated on one side, the other side must also be Illuminated. If, however, the edge is convex, if one side is Illuminated, the other side might be Illuminated, or Shadow Projected or Self-Shadowed, and so on. In practice, adding the illumination information increases the number of legal corner models to 3,256. Since Waltz assumed that each scene would be made up of blocks on a horizontal table top, any line segment separating the background (table) from the rest of the scene can only be labelled in one of seven ways. This fact reduces the number of corner models that can be used to label junctions on the scene/background boundary to 245. In other words, the scene-background boundary provides additional constraint.

2. The other improvement introduced by Waltz was a new method of searching through the candidate corner models. The method converges quickly on the possible interpretation of a scene. In general terms, the method is analogous to building a jig-saw puzzle. Just as one starts by assembling the edge pieces of a puzzle, the labelling process begins by labelling the scene/background boundary. In turn, this labelling constrains the labelling of internal edges due to the rule that in the case of planar objects an edge cannot change its type along its length.

The search activity actually comprises two stages, namely the use of selection rules to eliminate as many labels as possible by, for example, starting with the scene/background boundary, and use of a filtering procedure, a method of quickly eliminating candidate labellings for internal edges by applying the rule about edge type consistency. Let's see how the filtering procedure works by taking a unit cube as example, as shown overleaf.


F
F1
F2
F3
F4
F5
F6
A1
A2
A3
A.
B6
B5
B4
B3
B1
B2
B.
C3
C2
C1
C.
E3
E2
E1
E.
D3
D2
D1
D.

Step 1.
Compare A and B for mutually exclusive junctions. Since there are no out-going arrows in A, we have no in-going arrows in B.
Eliminate B1 and B6

Step 2.
Compare remains of B, viz. B2 B3 B4 B5 with C. Since there are no in-going arrows in C, eliminate out-going arrows in B.
Eliminate B5

Now there are no + labels in B, so
Eliminate C3 and Eliminate A3.

Step 3.
Compare remains of C, viz. C1 and C2, with D. Since there are no + labels or out-going arrow labels in C, there can be no + labels nor in-going arrows in D, so
Eliminate D1, D5 and D6.

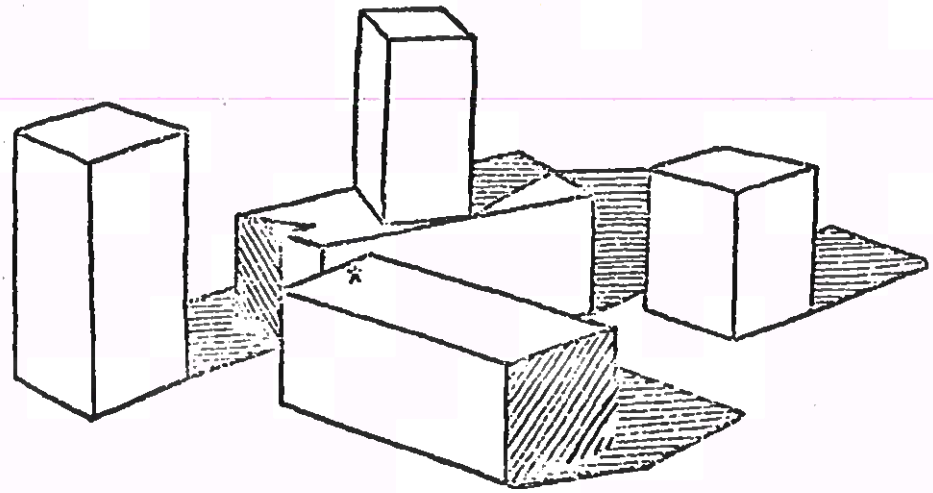
Step 4.
Compare remains of D, viz. D2, D3 and D4, with E. Since there are no + labels or out-going arrows in D, there can be no + or in-going arrow labels in E, so
Eliminate E3.

Step 5.
Compare E1 and E2 with F. Since there are no + nor out-going arrow labels in E, there can be no + nor in-going arrow labels in F, so
Eliminate F1, F5 and F6.

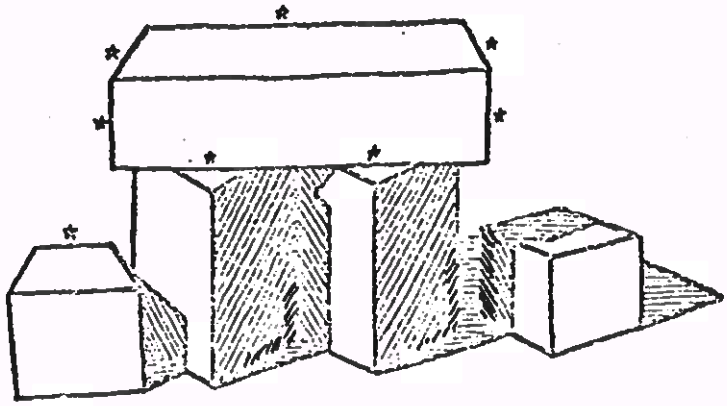
Step 6.
Compare remains of F with A. No further elimination, so filtering is complete.

In the case of a complex scene, the system might not be able to label every edge uniquely. So it is equipped with special case heuristic rules (rules of thumb) which try to find a plausible interpretation. For example, one heuristic eliminates interpretations that involve concave objects in favour of those that involve convex objects, and another prefers interpretations which have the smallest number of objects (this heuristic prefers a shadow interpretation for an ambiguous region to the interpretation of the region as a piece of an object). Also, special case heuristics deal with the labelling of non-trihedral vertices, the accidental alignment of edges, and missing lines in the picture description.

The program reached the stage where it successfully handles scenes such as those shown opposite. The segments which remain ambiguous after its operation are marked with stars.



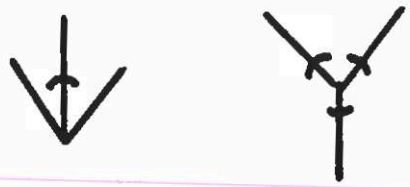
(39 SECONDS)



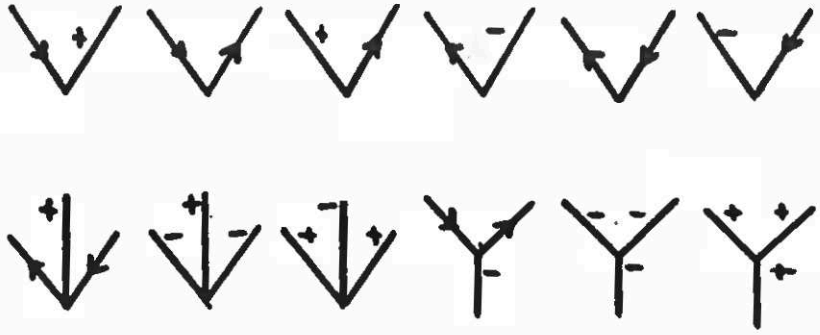
(48 SECONDS)

We are now in a position to understand why Guzman's program works. You will remember that we noticed thaat it worked best on scenes with convex trihedral vertices, that is with convex objects. Accordingly, we can eliminate from the set of Huffman's corner models all corners with concave edges, including those for the L that imply a hidden concave edge, leaving the set shown overleaf. Notice that L, FORK and ARROW junctions now have unique corner interpretations, where the + labels, which indicate convex edges, also match Guzman's links, i.e. we can derive Guzman's links by planting a link at a convex edge and no link at an occluding edge.

Also, link suppression rules (no link is placed across a line at a FORK junction if its other end is a barb of an ARROW, a leg of an L, or the crossbar of a T) are equivalent to the rule that the opposite ends of a line must have the same labelling. Indeed, the accumulation of link evidence based on the existence of two links between surfaces means in effect that both ends of an edge must agree that it is convex for it to be so taken. If only one end says so, i.e. one link, there is a conflict which must be heuristically resolved in Guzman's system.



GUZMAN.

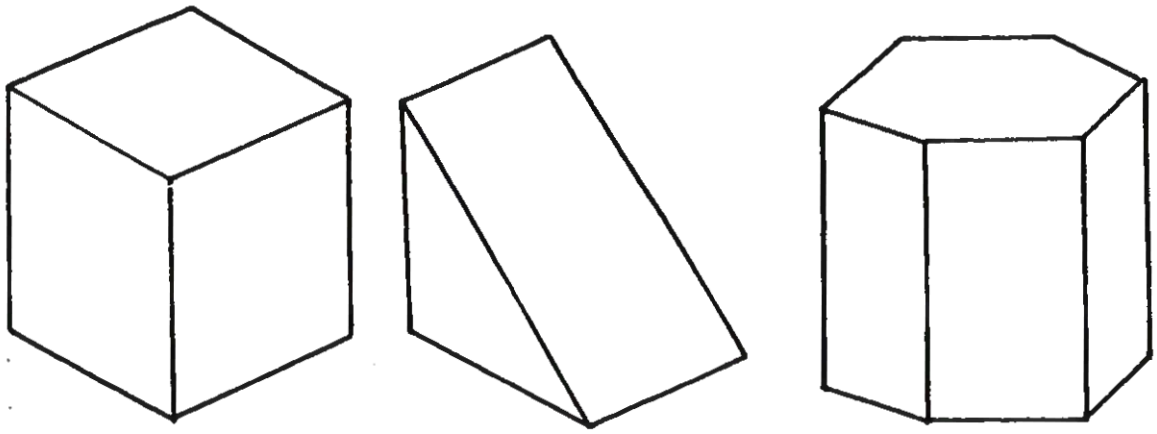


HUFFMAN.

KNOWLEDGE DRIVEN SEGMENTATION

We turn now to consider a program written by Roberts which takes a 2-D description of a scene and interprets it as representing a collection of 3-D bodies. In carrying out its analysis, the program uses three different kinds of knowledge.

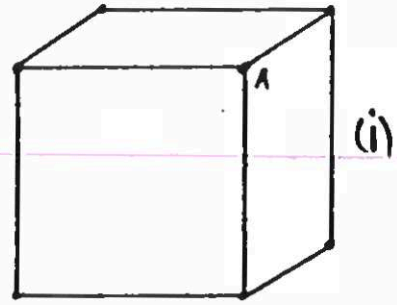
First, the program contains descriptions of three kinds of 3-D bodies, namely, a cube, a right-angled wedge and a hexagonal prism.



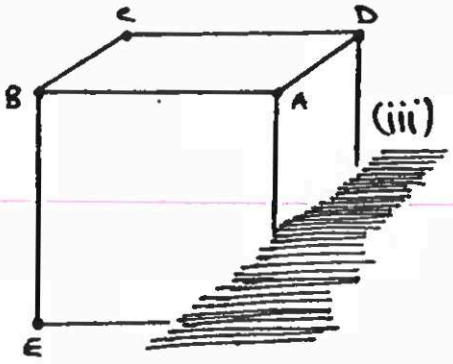
The assumption is that objects in the scene will be built from these bodies. In the program, these bodies, or prototypes as Roberts calls them, are represented in three-dimensions.

Second, the program extracts certain configurations of elements from the 2-D representation of the scene. These configurations are used as cues or clues in the process of finding out which bodies are present in the scene. The actual configurations are formed out of what Roberts calls "approved polygons". These are planar regions in the 2-D representations of the scene which could correspond to the surfaces of the three prototypes. Thus a triangle is an approved polygon because it could represent a face of the right-angled wedge; quadrilaterals (a quadrilateral is a plane bounded by four edges) and hexagons are also approved polygons. In the domain in which Robert's program works no other configurations are approved. In fact, what the program prefers as a cue is a combination of approved polygons and, ideally, combinations in the following order of preference:

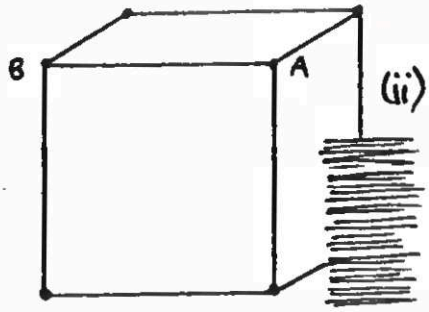
(i) Three approved polygons surrounding a point is the most informative cue combination. For example, overleaf, we see three quadrilaterals with a common point A. This cue in the 2-D domain points to the cube prototype in the 3-D domain.



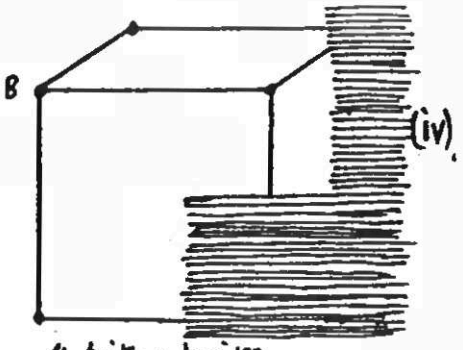
7 picture points



5 picture points



6 picture points



4 picture points

(ii) If the program is not able to find three approved polygons surrounding a point, it looks for the somewhat weaker cue of two approved polygons sharing a common line, for example, line AB.

(iii) If the program cannot find any cues of types (i) and (ii), it will accept the still weaker cue of a single approved polygon with a line coming from one vertex, for example ABCD, with line BE.

(iv) Finally, if the program is not able to find an approved polygon with a dangling line, it will look for a single point from which three lines emerge, for example point B.

What the program does is to use the best 2-D fragment to select a 3-D prototype. Roberts uses a predetermined order of prototypes (cube-wedge-hexagonal-prism) over which the program searches for a prototype fragment to correspond to the 2-D fragment. That is, given a set of 2-D points

forming a particular relationship, it searches the prototypes looking for a set of points which have a similar relationship. Making comparisons on the basis of similar point arrangement is a kind of topological comparison: it makes no assumptions about the relative size of body and prototype. A size match would only occur in special situations where the body was identical to the standard prototype. Only then would there be an exact match between the 2-D points projected by that body and the model points with which they have been paired. Normally there would be a numerical mismatch between 2-D points and prototype points, suggesting that the prototype will need to be altered to match the input data.

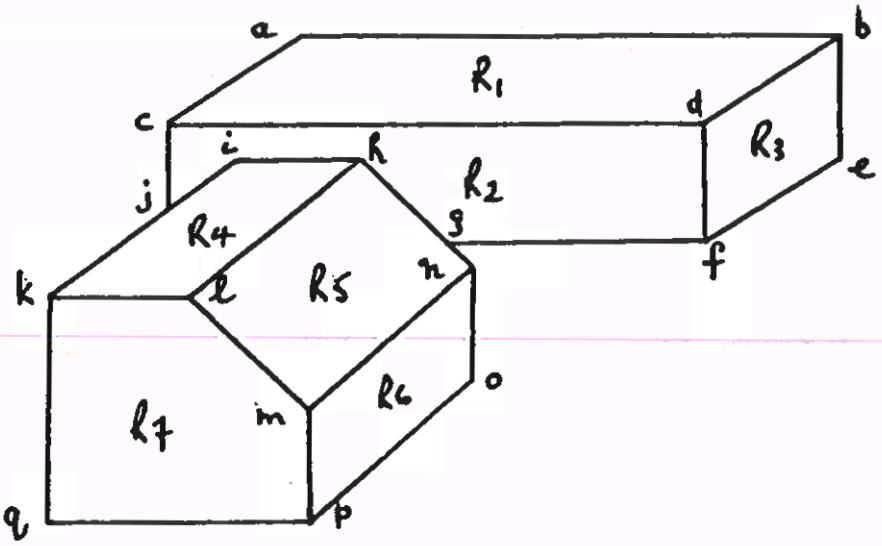
As a third form of knowledge, the program needs to know how to stretch, rotate and project the 3-D prototypes so that it can make this kind of match. It does this by solving a series of simultaneous equations. Although the method of doing this will not be described here, it is equivalent to two transforming operations on the prototypes. One operation stretches and rotates the prototype to fit the cue configurations. The other utilizes knowledge about projective geometry to check that a 2-D projection of the 3-D prototype could fit on to the appropriate part of the 2-D scene description. In the latter case, three possibilities arise:

(a) A fit means that the program has found the correct prototype and the correct transformation.

(b) If some of the prototype's points fall outside the points in the 2-D representation of the scene, this means it has selected the wrong model.

(c) If all the prototype's points fall inside the points in the 2-D representation but do not account for all, this indicates that the scene contains a composite body, made up of more than one prototype. The program has to decompose the composite body into sub-parts that can be checked out as transformed prototypes.

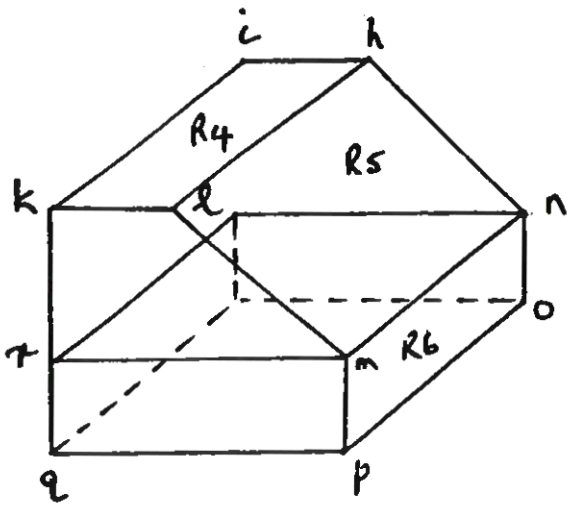
The best way to understand Roberts' program is to work through an example of a particular scene, bearing in mind the kinds of knowledge that the program brings to the task. Consider the scene shown next, where each line corresponds to a visible edge in the scene. We begin by naming the regions R1, R2, etc. Although there are seven regions, only five are approved polygons, namely R1, R3, R4, R5 and R6 which are four sided. R7, which is five-sided, and R2, which is seven-sided, are not approved polygons.



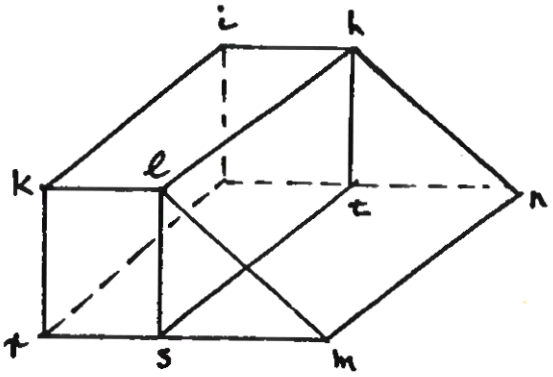
As a first step, the program examines the 2-D description looking for a point surrounded by three approved polygons. Since there are no instances of this combination, it looks for two approved polygons which share a common line. There are three instances, namely, R1 and R3 which share line bd, R4 and R5 which share the line hl, and R5 and R6 which share the line mn. Suppose it takes the R4/R5 combination. The program will find that a cube and perhaps other prototypes have two approved polygons which share a common line. So it picks a line in the cube prototype which has the approved polygons around it. Next, it picks a polygon from both the 3-D cube prototype and the 2-D scene description as starting points, and proceeds to list topologically equivalent point pairs. When finished, it has a list of six three-dimensional points from the prototype and a corresponding list of six two-dimensional points from the 2-D representation of the scene. Now its task is to transform the three-dimensional prototype fragment to match the two-dimensional input fragment. Thereafter, it calculates the overall fit between prototype and 2-D description to decide if the prototype chosen is the correct one. In the case of R4 and R5, the transformed cube prototype does not fit the 2-D data sufficiently well. The same is true in the case of the R5/R6 combination. However, the cube prototype can be transformed to fit the 2-D data giving rise to the R1/R3 combination. The existence of the two T-junctions j and g, which can be joined via c, d and f is also used as contributory evidence for the interposition of one block in front of another one. Once the choice of prototype has been confirmed by the goodness of the fit, the program uses the prototype to supply information about the position of unseen lines in the scene, and enters them into its final description of the scene.

The program still has to account for the body in the foreground of the scene. Since the R4/R5 and R5/R6 combinations were unsatisfactory cues, it looks for the next kind of cue, namely an approved polygon with one dangling line. Notice that there are two, R4 with line kq, and R6 with line

pq. The latter combination invokes a cube prototype, but this time the prototype can be rotated, stretched and transformed to fit the input data. Once again the program uses this prototype to supply information about the position of unseen lines in the scene, and enters them into the final description which is represented below:



By a similar process, the other combination R4 (now with line kr) also invokes the cube model which is successfully transformed to match the input data, leaving only the body shown below (h, l, m, n, t, s) to be identified. Again, the two approved polygons which share a line invoke the wedge model which fits the input data when rotated, stretched and transformed.



Before moving on, a few words of comparison between Guzman's and Robert's program might be useful.

1. Guzman's program segmented the scene called 'BRIDGE', into 8 separate bodies, namely:

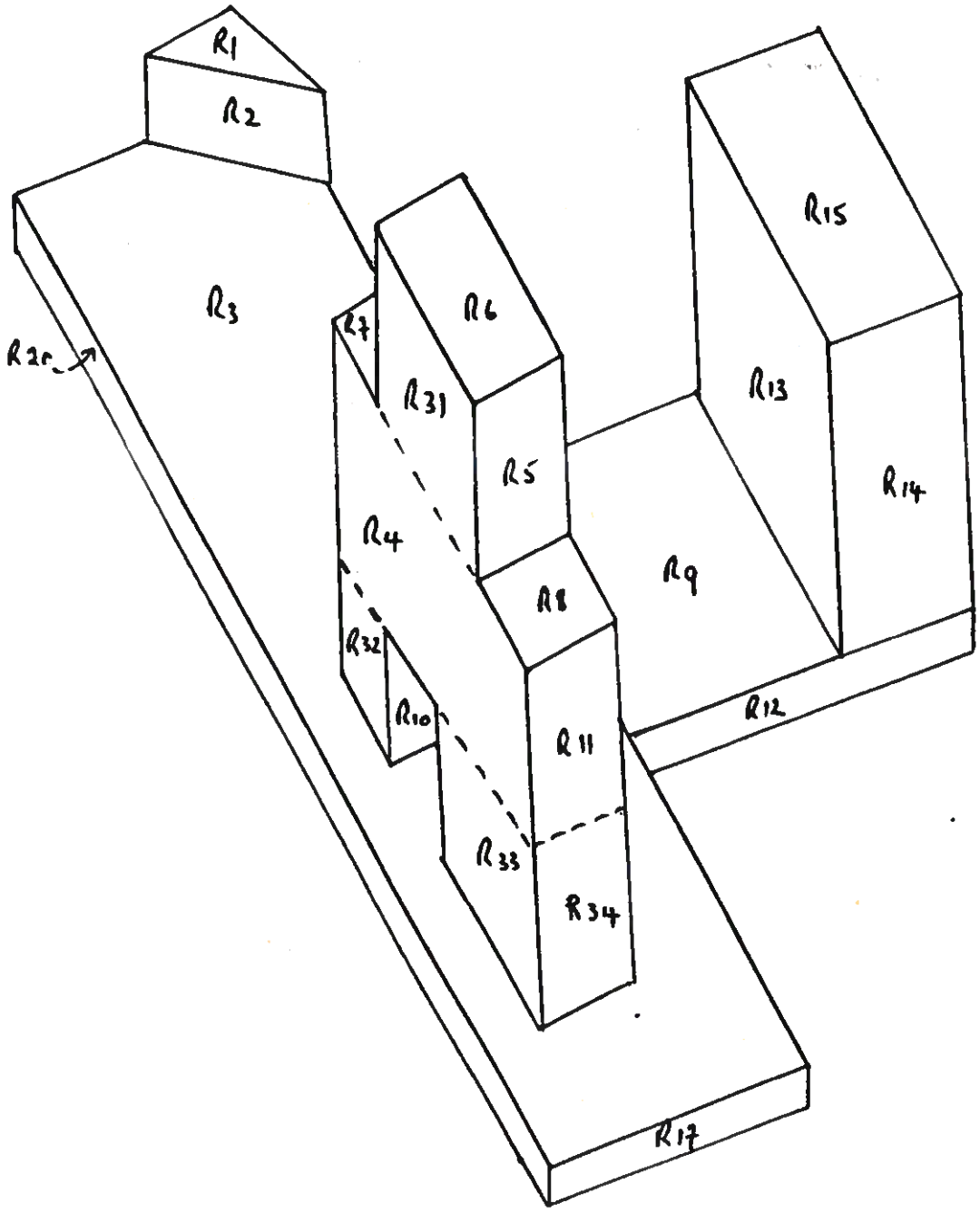
```
Body 1 : (R24 R9  R21 R27 R12 R25)
Body 2 : (R22 R26 R23)
Body 3 : (R17 R3  R20)
Body 4 : (R1  R2)
Body 5 : (R13 R14 R15)
Body 6 : (R19 R18 R16)
Body 7 : (R29 R28)
Body 8 : (R8  R11 R5  R6  R4  R10  R7)
```

The question is how would Roberts' program cope with this scene? We might expect it to arrive at the following conclusions:

```
(R24 R9  R21 R27 R12 R25) is instance of cube cf.  Body 1 above
(R22 R23 R26)             is instance of cube cf.  Body 2 above
(R17 R3  R20)             is instance of cube cf.  Body 3 above
(R1  R2)                  is instance of wedge cf. Body 4 above
(R3  R14 R15)             is instance of cube cf.  Body 5 above
(R16 R18 R19) is instance of cube (or wedge)  cf.  Body 6 above
(R28 R29)     is instance of cube (or wedge)  cf.  Body 7 above
```

So far, Roberts' program has made the same segmentation of the scene. However, at this point its analysis differs. Guzman's "Body 8" is not an instance of one of Roberts' prototypes. Instead, Roberts' program would decompose it into its primitive parts, as shown opposite, yielding:

```
(R10 R32)   is instance of cube   cf. Body 8 above
(R33 R34)   is instance of cube   cf. Body 8 above
(R4  R11)   is instance of cube   cf. Body 8 above
(R6 R5 R31) is instance of cube   cf. Body 8 above
```


R1
R2
R3
R2r
R7
R6
R31
R5
R4
R32
R10
R8
R11
R33
R34
R15
R13
R14
R9
R12
R17

So Roberts' program finds three more bodies than Guzman's program, i.e. missing edge data does not matter provided the outer boundary is intact. In contrast, Guzman's program is highly susceptible to missing edge information. The reason for this difference is that Roberts' prototypes carry with them information about 3-D structure whereas Guzman's corner models are derived from the 2-D appearance of a 3-D scene, and do not carry information about 3-D structure.

2. Notice that Roberts' first test, namely finding a point surrounded by three approved polygons, corresponds to Guzman's FORK heuristic. Notice also that his second test, namely find a line flanked by two approved polygons, is Guzman's ARROW rule. Finally, notice also Roberts' use of T-joints to provide evidence of interposition of one body in front of another one.

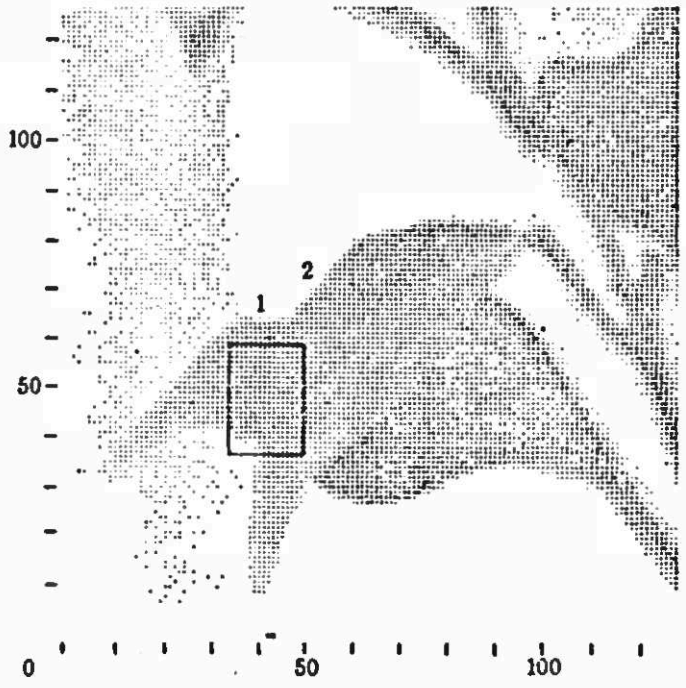
3. Although we discussed Guzman's program before dealing with Roberts' program, in fact Roberts' program was written about 4 years before Guzman's program. Although it doesn't identify objects, like the arch object discussed above, it does identify all the primitive bodies. e.g. cubes, wedges and hexagonal prisms, and can name them if required. Because of this it is referred to as a recognition program, and is cited by many as an important example of the theory of seeing which is based on the notion of a stimulus fragment invoking a prototype model. But in reality, Roberts' program is special case segmentation program because it analyses a scene into its constituent bodies, i.e. blocks, wedges and prisms. It does not recognize objects made from these components, such as arches, bridges, tables and so on.

TWO EYES

Segmenting natural objects

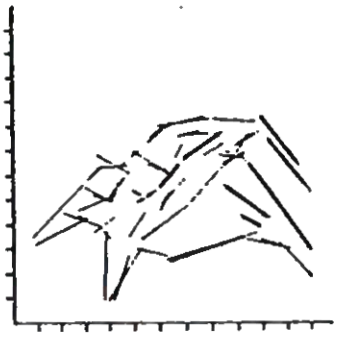
So far we have confined our discussion to the problem of segmenting scenes containing regular shapes. What about more complex, natural objects? Can these be handled using similar techniques?

Suppose we are interested in the problem of analysing a photograph of part of a plant. Plants are a challenging subject for analysis because their natural curved surfaces are difficult to describe using the precise mathematical methods that have proved satisfactory in more geometrical domains. Below, we see a digitised image of part of one of McLennan's plants. The actual intensity values that occur within the superimposed rectangle are given in Table 1 (see Appendix).



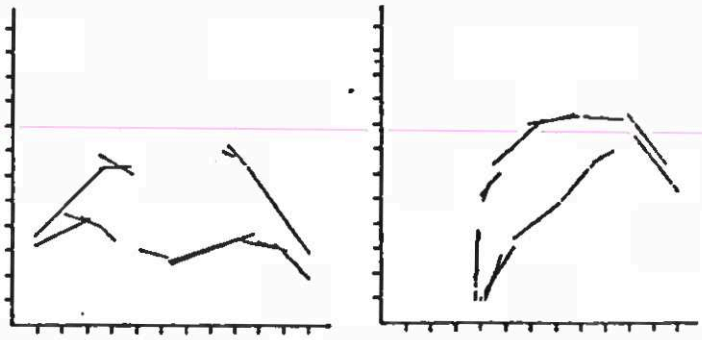
The image was processed by Marr's system in the way described previously. The contents of the data-base are drawn out below. The question is can Marr's grouping methods separate the leaves to achieve a satisfactory segmentation of the scene? Apparently they cannot handle this task.

The primal sketch does not contain enough information to separate the two leaves due to the lack of contrast across the edges of the leaf, as shown by the image values in Table 1. So the aggregation techniques deliver the form shown below:



To segment this form into the two components shown below, the system has to be given additional edge information. The need to provide this extra knowledge represents a defeat for Marr's view, and suggests that

building richer descriptions is not the solution to the problem.



But before we concede victory to the view that the analysis is controlled by high level knowledge, let's consider the assumptions on which it rests. The basic assumption is that the visual system is able to extract some distinctive features which suggest plants as context. But what are these distinctive features which act as cues? In the case of plants, some typical cues might be the simple shapes, PEAK and BAR, where PEAK suggests a leaf tip and BAR suggest a stem. Similarly, NOTCHES formed between BARS suggest branching in stems.

Although no one has implemented a leaf recognising program, it might include models of plants comprising characteristic plant parts and relationships between them. For example, a PLANT has such parts as STEM, LEAF, NODE, ROOT-NODE, MAIN-STEM; and the characteristic relations include facts about plant structure, for example every leaf is-supported-by a unique stem. The plant parts might be models themselves, for example the leaf has parts, TIP, VEINS, BASE, MARGINS, and these have relations like symmetry relative to the MID-VEIN. Given the existence of these plant models, and we beg the question how they were acquired, computing suitable cues for invoking them would not be a difficult task. However, invoking a model is only part of the process: the choice of model has to be verified. In the case of Roberts' program, this was achieved by stretching, rotating and transposing the prototype to fit the image data, and to provide missing edge information. To provide the missing edge information to segment the plant specimen shown above, the leaf recogniser would need to be able to stretch, rotate, twist and transform its prototype to match the input data, an extremely difficult matching problem.

Besides the difficulty in making the match between prototype and specimen, an obvious counter argument is that such a program would need to have the kind of specialized knowledge about plants which only a botanist

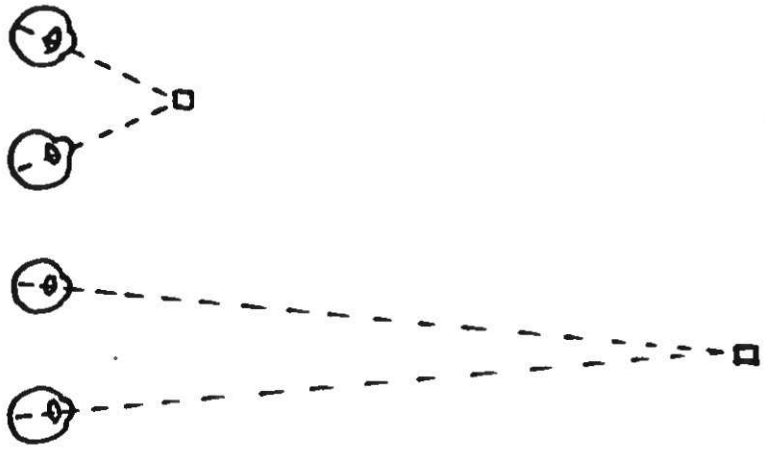
has. Yet people can recognize part of plants without this kind of detailed knowledge, so it should not be necessary to bring so much knowledge to bear to partition such a scene.

But human beings have two eyes, not one. The extra information available from a comparison of the descriptions generated by two eyes might solve the problem. We will turn our attention to this next.

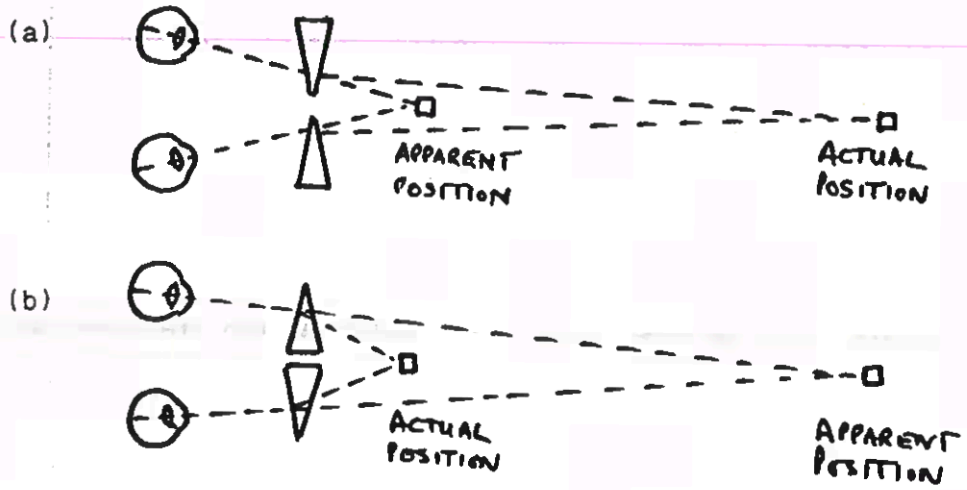
Seeing depth

Up till now, we have been discussing visual processing in the context of a single eye. But we have two eyes which work together to provide additional information about the properties of bodies in a scene.

Whenever we look at a body, our eyes pivot and alter their focus so that their images are projected clearly on to both foves. This pivoting of the eyes is known as convergence, and the amount that the eyes have to be converged is signalled to the brain to provide information about how far away the body is from the viewer. For example, the diagram below shows how the eyes pivot inwards for viewing near bodies, and outwards for viewing distant bodies.



A simple experiment shows that the convergence angle is used directly to signal distance. What happens if a pair of prisms of suitable angle are introduced to bend the light entering the eyes, so that they have to change their convergence to bring the images on to the centres of the foveas? If the prisms are placed to increase the angle of convergence, as shown in (a), bodies will appear far and small, whereas if placed to decrease the angle of convergence, as shown in (b), bodies will appear near and large.



So the difference in convergence can provide information about the relative position of bodies in the third (depth) dimension, to enable a scene to be segmented.

We can see that the convergence mechanism is analogous to a range-finder. But there is a serious limitation to range-finders: they can only indicate the distance of one body at a time, namely the body whose images are brought into correspondence in the two eyes by the convergence mechanism. When many objects are present in a scene, a different strategy is required.

Because the eyes are separated (by about 2 1/2"), each retina receives a somewhat different view of a scene. This can be appreciated quite readily by fixating a near body, with first the right eye closed and then the left eye closed. It will appear to shift sideways in relation to more distant bodies, and to rotate, when each eye receives its view. The slight difference between the images is known as 'disparity', and this is the basis for stereoscopic vision.

We can experiment with this, using a device called a stereoscope which was invented by Wheatstone in 1833. It presents any two pictures separately to the two eyes. Normally these pictures are stereo pairs, made

with a pair of cameras separated by the distance between the eyes, to give the disparity which the brain uses to give stereo vision.

Clearly the ability to fuse images on corresponding points of the two retinae (as in convergence) is a remarkable property of the visual system. Fusion from non-corresponding points (as in stereo vision) is even more extraordinary. Do we have any explanation about how fusion is achieved?

One rather obvious explanation is based on the notion that the binocular fusion system might work by matching up distinctive features in the two separate stereo images, 'interpreting' the disparities as depth information. But while it is true that the degree of similarity between the two parts of a stereogram is extremely important in achieving fusion, Julesz has shown that stereoscopic perception can occur in the absence of patterns or contour information.

In an important series of experiments, Julesz demonstrated that disparity of elements alone is a sufficient stimulus for the depth perception of dot patterns. He used pairs of computer-generated dot patterns, each containing about 10,000 elements. When identical copies were presented, one to each eye, they appeared quite flat i.e. two-dimensional. However, when a square array of elements in the centre of the right-hand member of the pair was displaced sideways by a distance of about four elements, this produced retinal disparity and the displaced section was seen lying a plane in front of or behind the remainder of the pattern depending on whether the lateral shift of the displaced section took place in the direction of the nose or ear of the observer.

Due to the dot pattern experiments we can be sure that any perceived 3-D structure (such as the central square of the stereogram) must occur at or after the point of fusion of the information from the two eyes, and no earlier. In turn this suggests that the binocular fusion system compares the fine-grain structures of the two monocular patterns, picking out those points of the two that are similar and fusing them, but discarding (or ignoring) mis-matches. Understanding how it does this is the stereoscopic matching problem

According to Marr and Poggio, stereo matching should take place between elements which are reliably related to surface markings and discontinuities. Clearly, for reasons given earlier, simple intensity changes are poor candidates. Raw primal sketch components might be better: in fact, Marr and Poggio use zero-crossings for matching purposes. For them, physical considerations impose three constraints on matching

(i) a pair of candidate edge elements to be matched must be physically similar if they have originated from the same place on an object's surface (the compatibility constraint).

(ii) any item in one image should match only one item in the other image (the uniqueness constraint).

(iii) disparity should vary smoothly almost everywhere in an image (the continuity constraint).

Marr and Poggio have proposed an algorithm for solving the stereoscopic matching problem. It has five main steps.

1. Left and right images are filtered, at a range of scales (just as if starting out to build a raw primal sketch).

2. Zero crossings are localized within these representations.

3. At coarse scale, matching takes place between pairs of zero crossings of the same type in the two images.

4. Once matches have taken place at coarse scale, the output is used to control a vergence system which changes the positions of the elements in the representations of left and right images to bring them into correspondence. In this way, the matching process gradually moves from dealing with large disparities at low resolution to dealing with small disparities at high resolution.

5. When a correspondence is achieved, the final step is to store the information in a buffer store, called the 2 1/2D-sketch. The reasons why it is called 2 1/2D instead of 3D is as follows. We begin the explanation by recollecting that zero crossings in the convolutions are caused by sharp changes in colour or reflectance of the surface, scratches on the surface, sharp changes in the shape of the surface, and so on. So, at best the stereo algorithm returns disparity values along some set of contours in the image. This means that depth surface orientation can only be explicity determined along such contours. To reconstruct a full 3-D description of the surfaces at all points in the image, the method would need to be extended. This is a current research problem.

So what is the value of stereoscopic vision? Julesz has suggested that the main reason for its emergence was to break the camouflage of a motionless prey (if it can do this, what effect would a binocular representation have on our leaf problem?). Whether or not this is true, binocular vision has enabled man to develop skills with his hands for which the ability to make very precise judgments of depth, in particular close judgment, is obviously very important. Before concluding this discussion of stereo vision, we should note that stereo is only one of many ways in which we see depth, and it only functions for comparatively near objects (up to about 20'), after which the differences between the images become so small that they become effectively identical.

Psychologists have discovered two additional sources of distance

information, namely static and dynamic cues. The former are mainly simple consequences of the geometry of the retinal image, and include relative size, perspective and interposition whereas the latter are the consequences of observer movement, and include motion parallax (where the image of a near object moves a greater distance across the retina than the image of a far object).

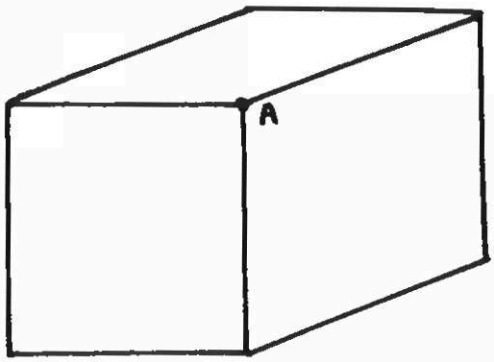
III. HIGH LEVEL ANALYSIS

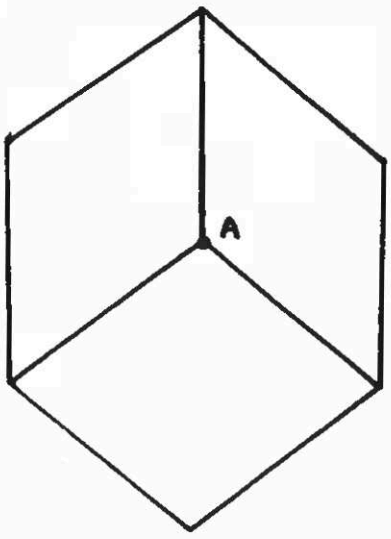
RECOGNISING 3-D OBJECTS

So far, we have considered visual processing as a hierarchical sequence of tasks, starting from the extraction of feature information from an image, followed by partitioning the data to yield evidence of the existence of separate bodies. Now, we have reached the top level in the hierarchy, namely recognising objects in a scene by matching properties of these bodies with descriptions (models) of objects stored in the computer's memory.

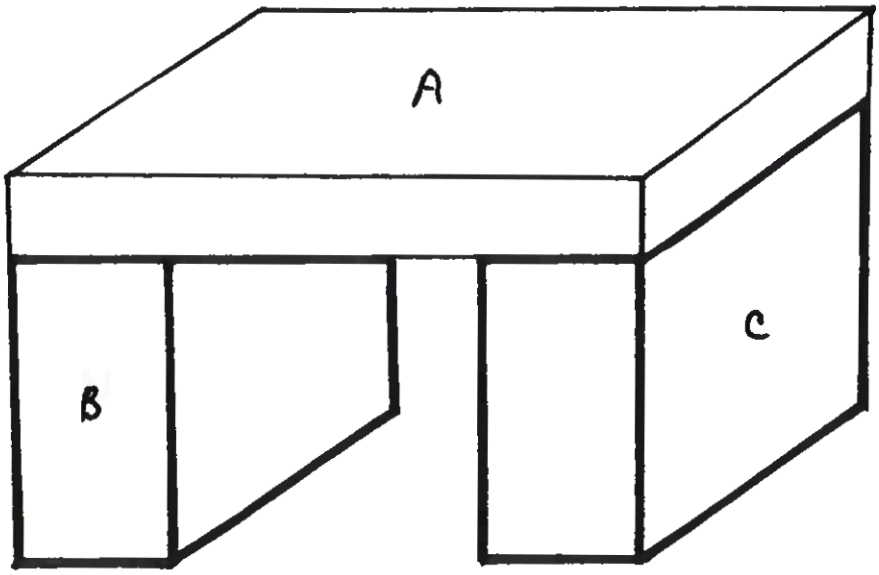
The first question is what form might these models take? Given that the shape of an object depends upon the viewing position (or in the case of a fixed viewing position, the shape of an object depends upon its orientation), it might be thought that the computer will need to record all possible object shapes in its model. Recognition would be achieved when the shape of the unknown body, or bodies, in the scene corresponds closely with one of the model shapes. The difficulty is that any recognition system equipped with many object models would have to store many thousands of views in its models. Besides burdening its memory, searching for the shape that matched the unknown body would be tedious, time-consuming and prone to error.

How might the models be made more compact? The answer is to try to build models which represent the invariant features of an object, i.e. features which do not change with viewing position. Consider a rectangular block viewed in two positions, as shown below:

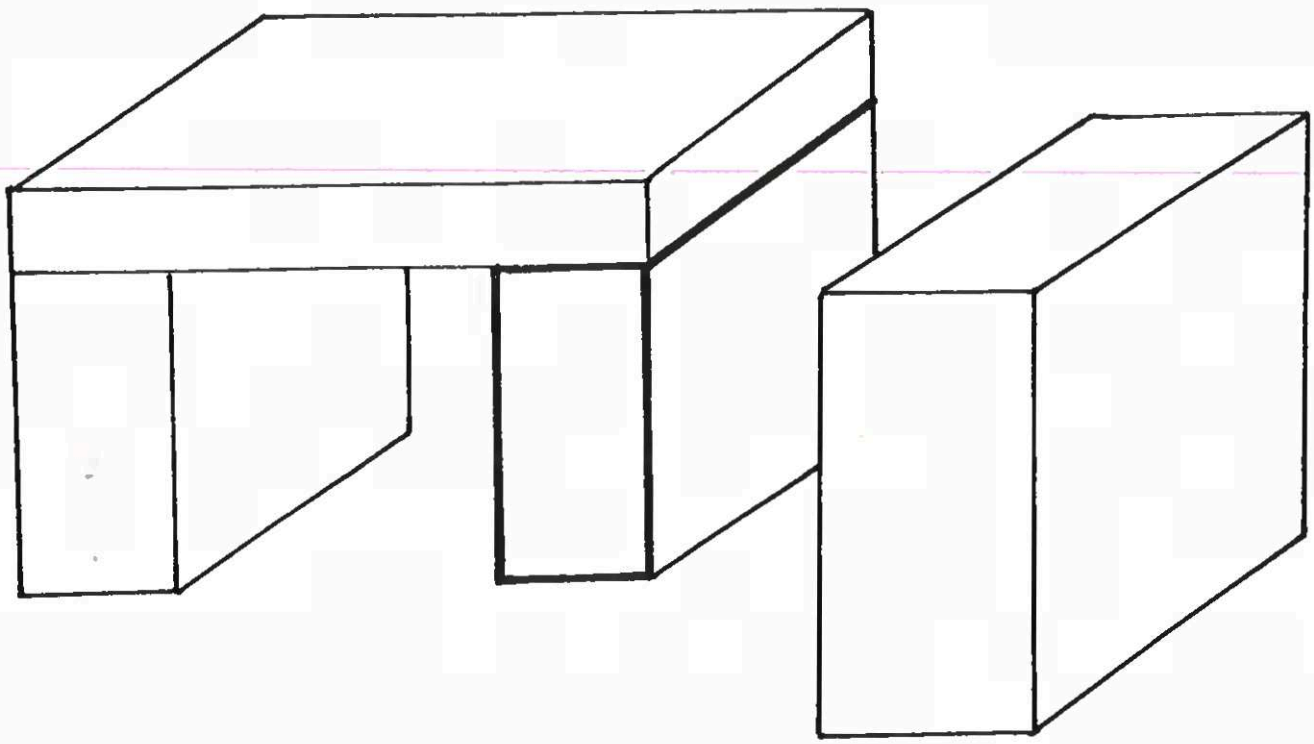




The edge information changes from one view to the other. What does not change is the shape of the faces and their relationships. Consider the vertex A, where the three faces intersect. In both views, point A is surrounded by three faces of the same type, namely three quadrilaterals (a quadrilateral is a plane bounded by four edges). So if the model of the block represents the block as three quadrilaterals bounding a point, any body in a segmented scene with these particular features will be seen as a block because the description derived by the low level processes will match with the model description. Of course, the situation is more complex: the 2-D appearance of a block will be affected by the presence of other bodies. If, instead of an isolated block, the object is an arch, as shown below, the two supporting blocks will be characterised differently. Now, we have two quadrilaterals sharing a common edge. So the model of a block must also contain this description, so that the body fragments in the arch scene will match the block model.



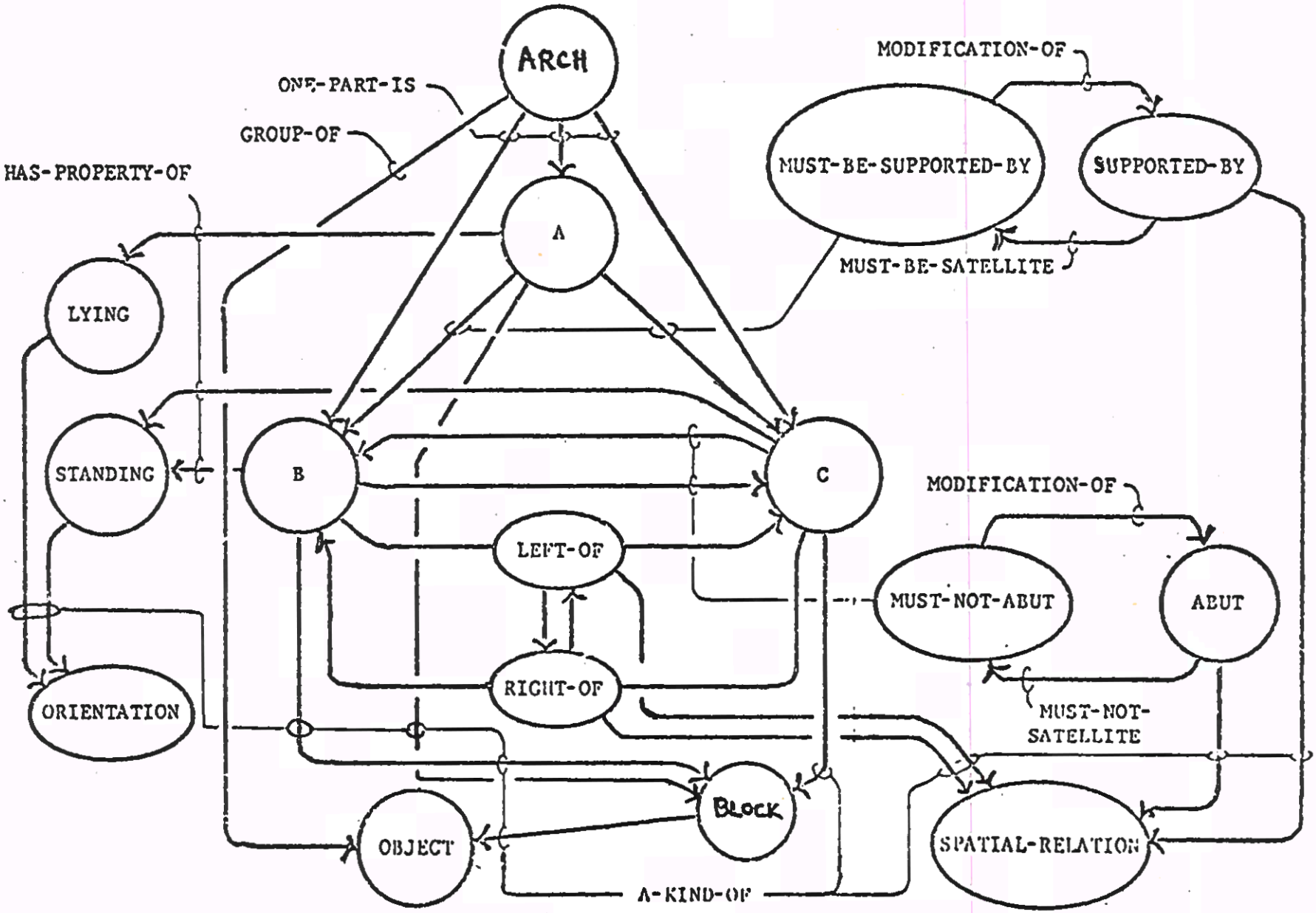
In scenes with multiple objects, a block may be occluded by another block, as shown next. So this means that the block model must respond to an even smaller picture fragment, namely a single quadrilateral with a dangling line.



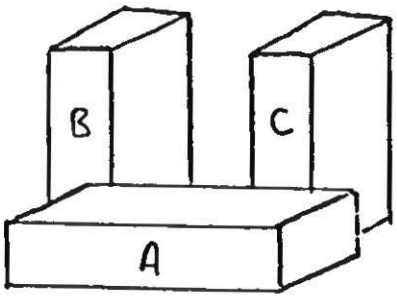
Notice that the evidence that an object is a block becomes progressively weaker, going from three quadrilaterals, to two quadrilaterals to one quadrilateral. If there are models of other kinds of blocks in the system, e.g. rectangular wedge, hexagonal prism, then the weaker body descriptions will also match these models (a rectangular wedge would be modelled as two quadrilaterals and a triangular face; a hexagonal prism as six quadrilaterals and a hexagonal face). In that case, the system would try to match the body description to the most likely model first of all.

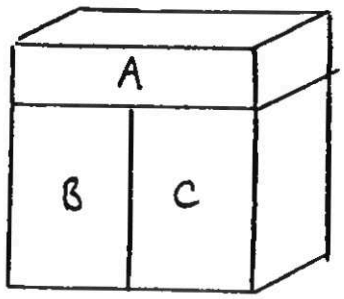
Suppose the system is presented with the view of an arch. Suppose, too, that the system has identified bodies A, B and C as rectangular blocks. This is but part of the story since the system does not know that there is an arch in the scene. Just as it needs models so that it can recognise bodies, it needs other models to enable it to recognise objects made out of these bodies. These models will be more complex than the models used so far.

So, how might an arch be represented? A convenient way of representing objects, parts and relationships is to use a directed graph, where bodies are represented by nodes and relationships are characterised by arcs linking nodes. Let's apply this technique to the arch structure. The graph is shown overleaf.

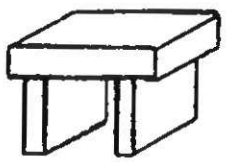

ARCH
ONE-PART-IS
GROUP-OF
HAS-PROPERTY-OF
MODIFICATION-OF
MUST-BE-SUPPORTED-BY
SUPPORTED-BY
MUST-BE-SATELLITE
A
LYING
STANDING
B
C
LEFT-OF
MODIFICATION-OF
MUST-NOT-ABUT
ABUT
RIGHT-OF
ORIENTATION
MUST-NOT-SATELLITE
BLOCK
OBJECT
SPATIAL-RELATION
A-KIND-OF

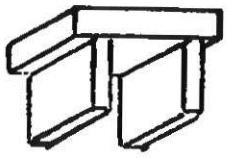
The top node is ARCH, which is decomposed into three nodes, A,B,C, each of which is (i) a-part-of ARCH, (ii) a-kind-of BLOCK, (iii) a-kind-of OBJECT. To distinguish an ARCH from other structures which are similar to, but not examples of, an ARCH, as below, we have to add additional information.





For example, block A must-be-supported by blocks B and C; block B must-not-abut block D, and so on. Note that this description does not refer to any numeric properties of the image of an object, such as the shape of the blocks. Since the object may appear in a scene viewed from any perspective, as below, numeric details would not be very useful aids to recognition.





The abstract description captures the essential features that should be invariant in any image of the object.

Given that the recognition system is equipped with relational descriptions as models of objects, the object recognition task can be seen as involving two steps:

a) build up a relational description of the unknown body, in terms of its components, their characteristics and their relationships;

b) compare this relational description with the set of stored relational descriptions until a good match is found.

We will duck the problem of how the system builds the relational description of the unknown object. Instead, we will focus on the second task, comparing the unknown description with the model description.

An object description is a graph structure, made out of nodes and relationships. A model description is a graph structure made out of nodes and relationships. So the problem is to determine when two graphs match, i.e. when they are isomorphic. But again this is an over-simplification since in the majority of cases the unknown object description will differ in certain respects from its model due to imperfect edge evidence, missing parts, extra parts, distortion due to the viewing perspective, and so on. In other words, the description of the unknown object will be weaker than it would be under ideal conditions. To restrict search time, and minimise the likelihood of an incorrect match, it is important for the system to restrict the number of candidate models used for any recognition instance. So rather than match an unknown description with each and every model, it is preferable to use an indexing method. Each model is equipped with an index of key features, such as the main components and their connectivity relationships. Given an index computed from an object in the image, a list of models with the same index is immediately available. Indeed several indices may be computed for a single model. Once the smallest set of candidates has been found, the actual comparison of descriptions can take place.

Matching an object description with a model description produces a list of similarities, and differences where they exist. For example, if the object description contains all the essential components and relationships stored in the model of an ARCH, the system will see the object as an example of an ARCH. If, however, a relationship is present that the ARCH model forbids e.g. contact between STANDING blocks, this difference will be noted and the match will be rejected. Similarly, if some essential model features are missing, the match will also be rejected unless the object is occluded. Under these circumstances, the absence of essential properties is tolerated.

Once the appropriate model has been selected, if it is equipped with appropriate numerical data, the program could discover, for example, if the object in the scene is a toy arch or life sized.

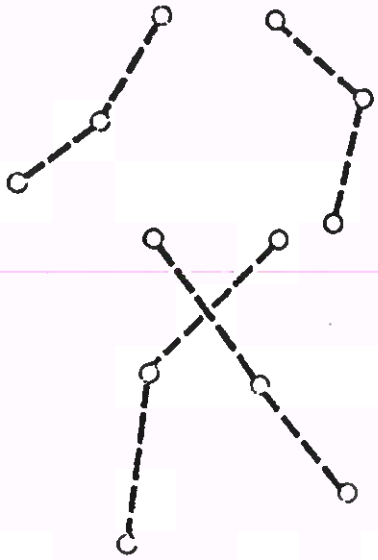
INFORMATION THROUGH MOVEMENT

So far we have considered static scenes and stationary observers. But more frequently bodies move, or the observer moves, or both, and the pattern of light at the eye distorts in a systematic continuous way.

We begin by recollecting that when we discussed receptive fields we noted in passing that they were sensitive to motion in particular directions. Although carried out on animals, the results of the experiments have been generalized to human vision, giving support to the concept of a visual information channel that is preferentially responsive to sideways motion. We must distinguish between two different kinds of sideways motion; real and apparent motion. Real motion is continuous displacement of a body from one location in space to another location at a particular velocity. If the observer's eye is stationary, a luminance discontinuity in the retinal projection will be displaced across adjacent receptors at the same angular velocity as those objects in motion. Apparent motion refers to circumstances in which motion is perceived when there is no continuous physical movement in the real world. For example, if two nearby stationary lights are alternately flashed, the observer will report seeing a single light rapidly moving back and forth. In this case, since there is no stimulus motion, the intervening receptors are not stimulated. Despite this, we perceive the light during its flight across the space between the two sources. This phenomenon is known as "phi", and is of course the basis for motion pictures.

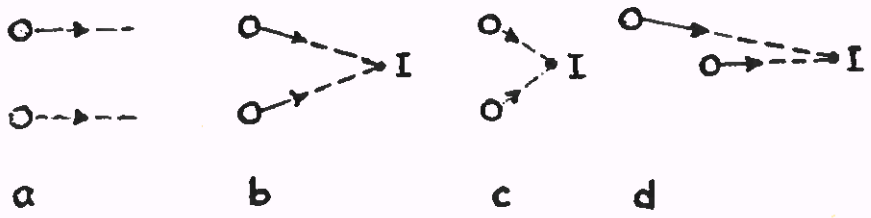
Consider a situation where light falls on a set of receptors at one instant, and at a nearby set in the next instant, and so forth, successively stimulating adjacent retinal elements. How does the visual system recover information about the body? Once again we are confronting the correspondence problem which we encountered when dealing with stereo vision. In this case, the correspondences are between similar features projected on to a single retina at different positions in successive time periods. If the visual system is able to identify corresponding points in successive time periods, it can specify the structure of the body and it can compute its velocity from the positional change information.

The question is how might the visual system solve the correspondence problem? In answering, we will consider a visual effect that was conceived by Johansson. He presented patterns of dots on a screen, similar to those shown overleaf (the links have been added to show relationships). Although the dots appear to be ambiguous, naive subjects can tell in a fraction of a second that they are seeing the movements of human figures. Not only are they able to distinguish between walking and jogging movements, but small anomalies like the simulation of a limp are also perceived.



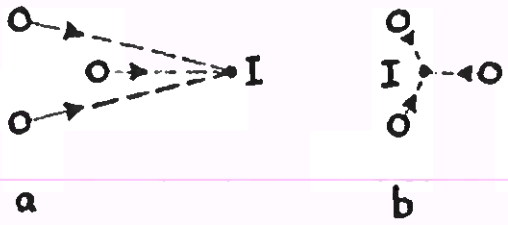
Johansson hypothesised the existence of a low level processing mechanism which extracts invariant relationships between the elements, presumably corresponding to those invariant relationships which we extract when we view people.

Besides discovering the visual effect, Johansson has also established a new grouping principle to add to those which we have discussed previously. We will call it the constant-distance-apart rule. To illustrate it, let us consider some simple configurations of dots, as shown here:



In (a) the stimulus consists of two bright dots in motion in parallel tracks and with the same constant velocity. They are seen as being rigidly connected and in motion along an approximately frontoparallel plane. Often the dots are reported as end points of an otherwise invisible stick or rod. In (b) the only difference from (a) is the direction of motion. Now the dots are converging. They are still seen as rigidly connected, forming a rod, and this rod is seen moving in a straight track which goes obliquely away from the observer at a specific angle to the frontal plane, i.e. it is seen as moving in 3-D compared to the 2-D motion seen in the first situation. In (a) and (b), the direction of the rod's perceived motion is at right angles to the motion of the dots on the plane. This is a consequence of the arrangement of the dots. Altering their relative positions, as shown in (c), does not influence the perceived direction of motion. Now an oblique rod is perceived moving in 3-D.

When a third (non-collinear) dot is introduced, a surface is perceived instead of a line. Given the dot arrangement shown in (a) overleaf, the observer always sees a rigid triangle moving obliquely backward in space.



In most cases, the surface is seen as having a frontoparallel direction in space. As in the previous examples, the point of convergence determines the perceived direction relative to the observer. For example, concurrent motion towards a point in the centre of the triangle results in a perceived motion of the triangle in a radial direction, as shown in (b).

How does the constant-distance-apart rule help us to interpret patterns of dots as human figures? Let us consider a simple case, namely motion of dots in 2-D without occlusion. Given a collection of dots which represent a structure, one possible algorithm would be as follows:

1. In frame 1, link together all the dots into a network 1, where the nodes depict dots and the arcs depict connectivity relationships, including length.

2. In frame 2, link together all the dots into a new network 2.

3. Compare networks 1 and 2, and remove the connectivity relationships between dots which have changed their relative positions from frame 1 to frame 2, i.e. have changed their length.

4. Compute velocity of movement, using positional changes of invariant features from frame 1 to frame 2.

Notice that the new description produced by comparing network 1 and network 2 is a structural description of the body in the scene. By applying the constant velocity assumption, it would be possible to compute positions of the dots in successive time frames.

Essentially the above strategy was used in a program written by Clocksin. Having generated structural descriptions, the program also recognised these descriptions as instances of human activities like walking or falling by comparing the derived descriptions with stored idealised descriptions of these activities. In carrying out this analysis, Clocksin's system had to

cope with some missing and extra body parts caused by some minor self-occlusions by body parts, e.g. hand obscuring light at waist.

Recovering structure in this way is analogous to depth perception through stereopsis, with successive frames substituting for adjacent images and displacement values playing the role of binocular disparity of elements.

How well does Clocksin's explanation confront the psychological data? While it handles single figures moving sideways on a plane, it is far from obvious how to generalize this approach to handle more complex problems. For example, take a more complex case where there is motion of two human figures on a plane. If they are dancing together, the system will generate a single network to represent both figures. The question is how will it segment the network into two sub-networks. Can it do so without invoking high-level knowledge about dancing?

To answer this question, we turn to work by Ullman at M.I.T. which abandons the planarity assumption. For example, if a transparent beach ball with tiny light bulbs mounted in randomly chosen positions on its surface is set spinning in a dark room, the correct spherical layout of the lights is seen immediately. When the spinning stops, so does the perception of the spherical array. The question is how does one see the correct 3-dimensional structure when very many 3-D structures might have produced the moving 2-D retinal projection? The answer is that the interpretation process must incorporate some internal constraints that rule out most of the possible 3-D interpretations, in favour of a unique solution. These constraints can be thought of as implicit assumptions about the physical world which, when satisfied, imply the correct solution. The constraint that Ullman proposes is called the rigidity assumption i.e. any set of elements undergoing a two-dimensional transformation which has a unique interpretation as a rigid body moving in space should be interpreted as such a body in motion. Notice that Ullman's rigidity assumption is similar to Johansson's observation that rigidity has a special role, as expressed in his "constant-distance-apart" rule.So, under the assumptions
1) that the correspondence problem has been solved, and
2) that the objects are rigid bodies,
Ullman has derived what he calls the structure-from-motion theorem. It states that three separate views of four non-coplanar points on a rigid object uniquely define the 3-D structure and motion of the object. The implementation of this theorem involves the following steps

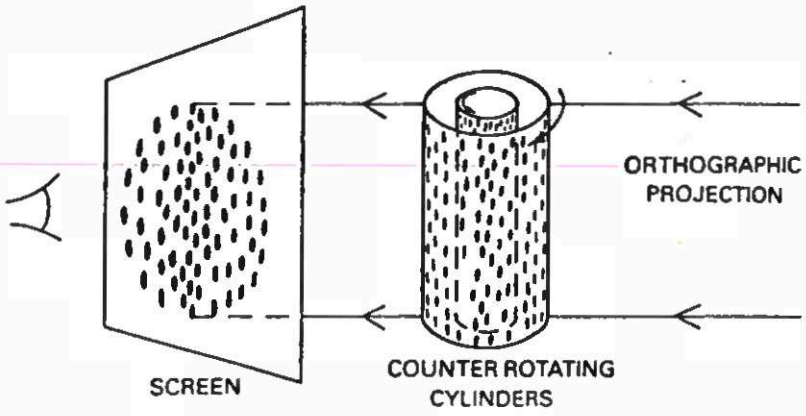
1. The image is divided into sets of four points

2. Each set is tested to see if it has a consistent rigid-body interpretation in the three views. In many cases, after this first form there will be at least one consistent set for each object in the image.

3. Each remaining point is tested to see if it belongs to one of the rigid body sets.

Ullman illustrates this through a more complex situation, one which involves two sets of points arranged in two non-planar configurations. It

comprises a projection of two co-axial cylinders on a display screen. Each cylinder is defined by 100 randomly chosen points lying on its surface. The common axis of the two cylinders is vertical.



Each eye's view appears as a near-random collection of dots. However, when the changing perception is viewed, the elements in motion across the screen are perceived as two rotating cylinders whose shape and angle of rotation are easily determined. Is there a connection between Ullman's work and Johansson's work? In Johansson's experiments, the correct 3-D structures are seen even when the snapshots do not contain four co-planar points. At best, in Johansson's snapshots only pairs of points are rigidly connected, such as the ankle and the knee or the knee and the hip. Rigid quadruplets of points just do not exist. This suggests that Ullman's method cannot be used to segment a network of dots, representing a couple dancing. A solution to this problem is still outstanding, as is a solution to the interpretation of dot patterns, representing figures which recede and approach. The latter might be handled if the constant-distance-apart rule is interpreted in terms of proportions of overall height of object rather than in absolute terms.

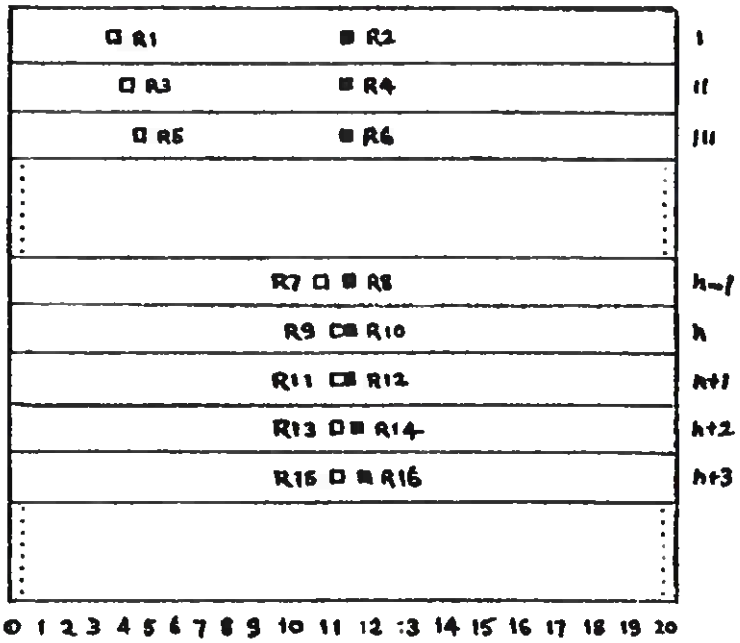
Perception of causality

We turn now to consider another example of a program that interprets a series of discrete images of objects that move in relation to one another. It was developed by Sylvia Weir. She was interested in the phenomenon of causality, first investigated by a Belgian psychologist called Michotte. What Michotte did was to show human subjects visual displays of 2-D coloured shapes such as squares, circles and triangles, which moved in relation to one another at different speeds. He asked his subjects to describe what they saw, and they reported impressions of objects, for example, billiard balls, chasing one another, pushing one another, passing by

one another, and fleeing from one another. Although Michotte claimed that different subjects interpreted a given kinetic event in the same way, subsequent investigators have found that people are less consistent than Michotte claimed.

Whereas Michotte argues that there is no question of an interpretation being superimposed on the impression of movement, rejecting the effect of past experience and an acquired knowledge of mechanisms, Weir suggests a knowledge-based explanation. Briefly, her view is that information extracted from a particular kinetic pattern invokes one of a set of memory models which describe actions like pushing, chasing and fleeing. Her evidence takes the form of a computer program which interprets a representative selection of the kinetic situations used by Michotte, and in a way which is consistent with Michotte's results, and those of subsequent investigators.

The program operates as follows. A kinetic pattern typical of the kind used by Michotte is discretely sampled to yield successive pictures representing successive instants in time, like the sequence of static pictures making up a cine-film. Conceptually, the input is shown below:



In practice, the initial grouping problem is avoided, and the program is given descriptions of the pictures rather than the pictures themselves. Obviously to compute motion, its first task is to detect changes which occur from frame to frame. It does this by entering into its data-base symbolic descriptions of the positions of elements in each frame. A symbolic description takes the following form

[at R1 R screen [frame 1] [colour black] [shape square] [position] [name]

Next, it matches the symbolic descriptions representing elements in pairs of successive frames to yield information about change of position. Making matches at this level involves handling the correspondence problem referred to above. That is, the program's task is to pair the descriptions of the same object from successive frames although that object has changed its position by an appreciable amount during the time interval between frames. To illustrate this process, let us see what the program makes of the change between frames 1 and 2. Comparison of the descriptions reveals that regions R2 and R4 have identical descriptions, which constitutes good evidence for pairing them. However, R1 and R3 have different descriptions on account of their differing positions, so they cannot be paired in a straightforward way. Inspection reveals that their colour is the same and differs from the colours of both R2 and R4. In the absence of any competitors, the most sensible pairing is deemed to be R1 with R3 and R2 with R4. So the program constructs a new description of the form

[A moves] [frames 2] [direction to-the-right] [speed 2] [from 1] [to 3]
[B stationary] [frames 2] [direction 0] [speed 0] [from 6] [to 6]

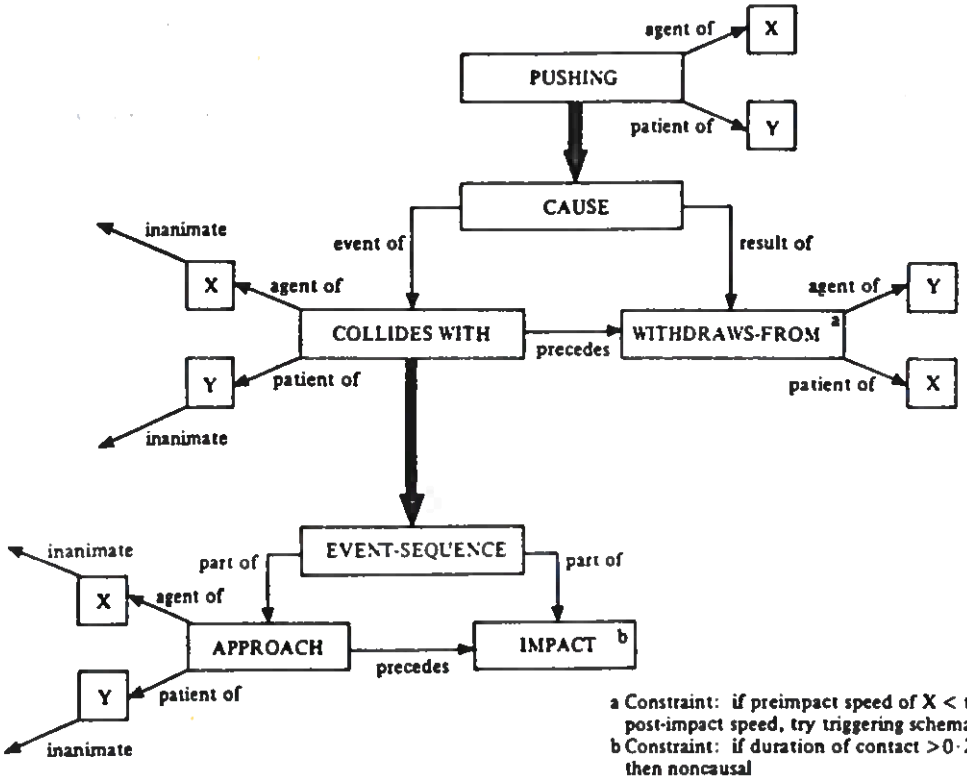
The successful pairing relies on the fact that few objects move in the environment i.e. there are no competitors to complicate the issue.

What the program tries to do now is to generate a symbolic description of the relationship between objects in the scene, such as the fact that a moving object A is approaching a stationary object B. Let us suppose that the experimental instruction [fixate midscreen] was input as part of the first frame's description. This instruction establishes the middle of the screen as the reference point for A's movement, and this is reinforced by the presence of object B as a target sitting at this reference point. Under these circumstances, adding the description [A moves] to the data-base results in a new description [A approaches B] [frames 2] [cue A moves] being generated.

When this description appears in the data-base, it activates a procedure called a demon. A demon's job is to look out for a particular set of circumstances. In this case, the demon is an impact demon which will be on the look out for A reaching B.

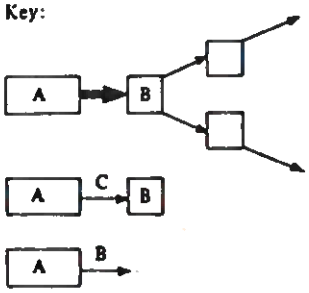
Meanwhile, having computed A's speed, the program introduces the assumption that objects move at constant velocity to enable it to predict its positions in successive frames. On this assumption, the expected next position of A is R5. Notice that besides simplifying the correspondence problem, this assumption also enables the program to detect changes in the speed of an object. In this case, however, the object is in the expected position, so its movement will not be perceived as a change. In fact there is a "no change" situation until Frame n-1 gives way to Frame n.

To understand what happens when Frame n is reached, we need to consider the program's knowledge about real actions, such as pushing, carrying along, launching. Descriptions of these actions, expressed in a network representation, are stored in the program's memory. For example, as shown below, the description for pushing includes such components as (X approaches Y), (X collides with Y) (Y withdraws from X).



a Constraint: if preimpact speed of X < twice post-impact speed, try triggering schema
b Constraint: if duration of contact >0·2 s, then noncausal

Key:



is to be read as: the node A can be viewed as node B and all the nodes which hang from it

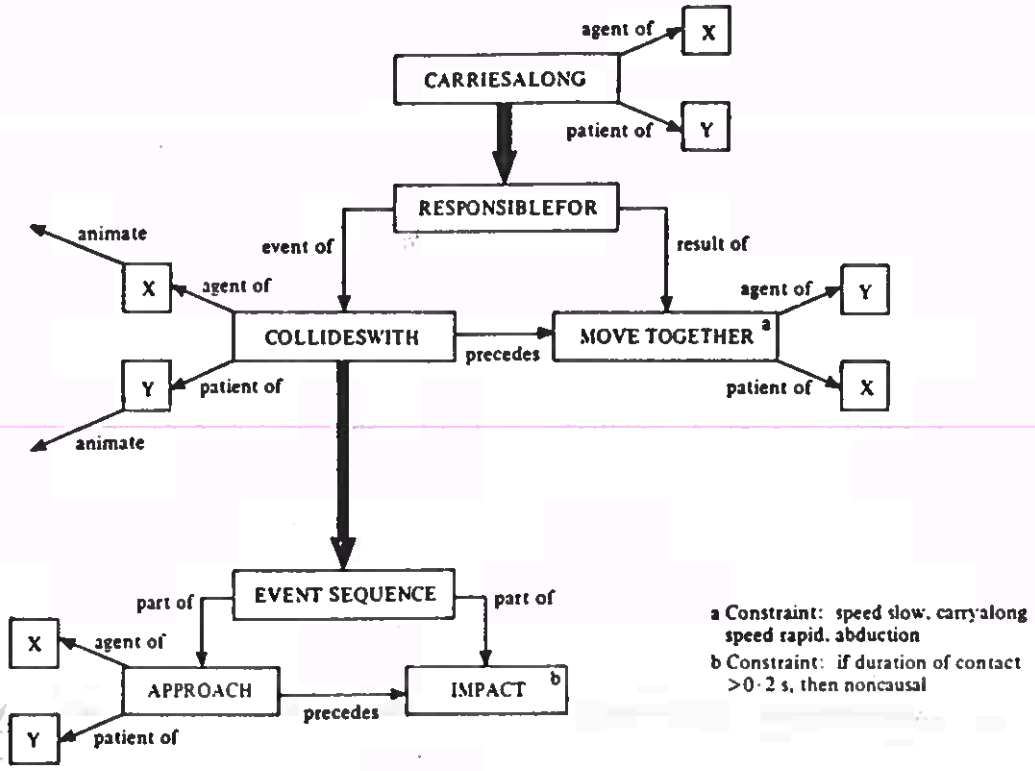
is to be read as: B (is the) C (of) A

is to be read as: A has the property B

The four instances of X and of Y denote the same individual: this notation was used to simplify the diagram by avoiding *identity* links between participants

Pushing schema.

(X approaches Y) is also a component of the carry along description, shown next.



Carry-along schema.

Both these descriptions were invoked when the lower level description (A approaches B) was entered in the data-base, and both predict that an impact will occur. This accounts for the fact that an impact demon was triggered to look out for A reaching B. When this happens, signalled by the appearance of the description (A next to B) in the data-base, the program infers that the expected collison has occurred and enters into the data-base the description (A collides with B). This description invokes another demon which has two tasks:

(i) it modifies the region-pairing process. The constant-velocity assumption is abandoned in favour of an absolute pairing between R9 (in Frame n) and R11 (in Frame n+1) with no sense of surprise. By pairing R11 (in Frame n+1) with R13 (in Frame n+2) and R12 (in Frame n+1) with R14 (in Frame n+2) the program establishes that B has started moving. This movement away from the midscreen reference point produces the description [B withdraws from MIDSCREEN]

(ii) it sets up a search for the consequences of the impact predicted by the stored description(s) invoked. As a result, when B starts moving, this movement will be interpreted as a consequence predicted by a stored description provided its time constraints are obeyed. Notice that when more than one stored description is a candidate, the time constraint information may be the critical factor in distinguishing which one should be rejected.

The program incorporates a number of subtleties which can account for much of the variety in behaviour displayed by subjects undertaking the psychological experiments. For example, being instructed to look at different places on the screen influences a person's perception of the kinetic event. Above we considered the situation in which the centre of the screen was the reference point. Suppose, instead, the reference point is the side of the screen. The program doesn't expect anything to happen at the centre. Instead, the screen itself is the reference frame, and adding [A moves] to the data-base description generates the description

[A movesacross screen] instead of [A approaches B]

This new description does not activate the impact demon so when Frame n is reached there is no reason why the region-pairing process should be altered. According to the rules invoked at Frame 2, R9 is expected to move to the right but R10 is expected to stay still. Consequently, at frame n+1 region R12 is paired with R9, and R11 with R10. But in the following frames, odd-numbered regions are paired together as before, as are even numbered regions. In other words, the picture sequence is interpreted as one object passing over another stationary object. The change in colour at Frame n+1 is noticed, but is treated as less significant than the fact that the movement is the expected one. Analogously, Michotte's subjects saw the passing effect under similar experimental conditions. Some also saw a small retreat of the stationary object as the moving object passes over it. This phenomenon is neatly explained by the pairing of R10 with R11 instead of with R12, as previously pointed out.

Just how good a computational metaphor is Weir's program? We have already seen that its interpretations of kinetic events are similar to the interpretations made by human subjects. But while the interpretations may be similar, are they produced in the same way? Just how well does the mechanism in Weir's program confront the psychological data? There are a number of weaknesses.

While the discrete sampling notion seems to be useful because it enables a system to judge change of speed by comparing actual changes of position with changes predicted by some built in rule, like the constant velocity rule, it introduces the correspondence problem. Matching the same elements in successive frames in a scene of any degree of complexity is a difficult task. At what level should the match be attempted? This question raises the kind of problems investigated by Lamontagne who was unable to solve the problem of whether to group feature points into a higher level unit before computing motion of that unit, or whether to compute the motion of individual feature points before grouping them into higher level unit(s). In the latter case, the correspondence problem is particularly difficult due to the large number of elements to be individually matched from frame-to-frame.

If we abandon discrete sampling and inferred movement information in favour of real movement information, can we avoid the need to invoke knowledge to account for the perception of causality? Put this way, the answer is likely to be "no, we do need knowledge about different kinds of motion so that we can communicate what we see to others". However, if we ask whether or not we need knowledge to see objects in motion, the answer might well be "no, we don't need knowledge to see something move". Unless we accept this latter statement, we have to face up to the logical problem of accounting for the acquisition of the knowledge used to mediate perception by a system whose operation depends on its existence.

Of course, the constructionists i.e. those who believe that seeing is model driven, might argue that the phenomenon of apparent movement, that is our ability to see an object in motion when the eye is stimulated with a succession of static pictures, indicates that the human visual system is able to infer the experience of motion. No-one would deny this: what is at issue is whether the visual system was deliberately designed to infer motion from static images shown in close temporal succession, or whether the phenomenon of apparent movement is a side-effect of a system designed to compute real motion. Of course, there could be a biological reason for having a system capable of analysing both kinds of motion. A predator stalking its victim over rough ground, through forest terrain and so on would provide intermittent information of its presence to its intended prey. Being able to take advantage of fragmenting information would be of considerable benefit.

What we may conclude, therefore, is that the low-level mechanism for identifying the elements in the Michotte displays and for computing their motion are implausible when the psychological evidence is confronted. However, notice the crucial distinction that arises between the ability to see bodies in motion and the ability to talk about these bodies as objects engaged in some particular form of activity. Although we might wish to discard the lower level mechanisms used by Weir, some notion like schema or model invocation is needed to account for the variability in the behaviour of Michotte's subjects. Clearly a physicist would have a richer set of kinetic analogies to draw upon to communicate his perceptions compared with a non-physicist, leading to a variety of interpretations of the same pattern of kinetic stimulation. However,it is quite another matter to be asked to concede that a physicist's perception of forms, shapes and movement patterns is entirely different from that of a non-physicist.

CONCLUSION

CONCLUSION.

We started from the position that seeing is so easy that it seems as if there is no problem to be explained. Through our efforts at exploring the design of artificial seeing systems, whether or not they operate in ways which are similar to the ways in which the human visual system operates, we have gained some insight into just how complex the underlying processes are. In the first lecture, we noted that the theory of seeing which is most popular in psychology and in artificial intelligence is the constructivist theory. To recap, its starting point is the compression of the 3-D visual world on to a flat 2-D pattern on the retina. Thus, there is never sufficient information contained in a retinal pattern to determine which 3-D scene accounted for that particular retinal pattern. But since our space perception of scenes is both valid and reliable, according to this theory we have to infer (Helmholtz's term) the third dimension. It is argued that we do this with the aid of additional information (called cues) contained in the retinal pattern, in combination with stored knowledge in memory which has been invoked to assist with the interpretation of the input data. According to this theory, therefore, seeing is a knowledge based process. The key issues are concerned with building rigorous explanations of the way in which knowledge is stored, invoked and used: no attention is paid to the issue of accounting for the acquisition and organising of this stored knowledge.

Unfortunately, one way in which the theory is inadequate is that it cannot explain how we can perceive bodies in a scene without invoking stored models. In the absence of a mental model, we might not be able to name an object; we might not know its function, but at least we ought to be able to acknowledge its presence. From studies of patients with severe brain injuries which render them blind, there is evidence that such people can actually see without knowing it. The patients in question undoubtably have injuries in the visual area of their brains; they never see a flashing light projected on the part of the retina associated with the brain damage; in fact they deny ever seeing anything there at all. But if a light is briefly flashed and if they are asked to <u>guess</u> its position by pointing to it, they can do so with remarkable accuracy. They can even guess whether a line flashed within the "blind" area is horizontal or vertical, even though they claim seeing no line at all, and find the whole exercise rather foolish. This phenomenon is known as "blindsight", and it is thought that this kind of perception is mediated by more primitive parts of the brain which were thought previously only to control eye movements.

We also noted the alternative explanation of seeing, advanced by the psychologist J.J. Gibson, to the effect that the succession of retinal images contains all the information needed to construct a 3-D representation of the visual world. Gibson's view is that this would be obvious to theorists if they did not think of the retinal image as a compressed or squashed two-dimensional picture, but rather as a source of organised optical information. His theoretical analysis suggests that the information content of the retinal image is rarely if ever incomplete. Therefore, the perceiver does not need to infer the third dimension, nor rely on past knowledge of what the scene might contain. All he has to do is to extract relevant information from an image to construct a 3-D representation of a

scene. Of course, it is difficult to reconcile Gibson's view with our own experiences. In particular, there is the problem of accounting for many of the practical phenomena which produce varying visual interpretations, such as the illusions, ambiguous shapes that we saw during the first lecture, or Michotte's kinetic patterns, and so on

We will conclude by asking whether there is any way in which these opposing points of view can be reconciled. The weakness of the constructivist theory is the inadequacy of the low-level information gathering mechanism, whereas the weakness of the theory of 'direct perception' is the lack of high-level interpretative mechanisms. Putting the two together would solve many of the problems thrown up by the separate explanations, and would enable us to hypothesise two different kinds of seeing, namely exploratory seeing and predicted seeing. The distinction between these two kinds of seeing is that exploratory seeing would be data driven for the purpose of building up memory models of objects and events, predicted seeing would be model driven in a goal context. Obviously, exploratory seeing would be a time-consuming process due to the large amount of information being handled by a system which is inherently slow (the nervous system). In contrast, predicted seeing would be a selective, hence fast, process, more in tune with a rapidly changing visual world.

We have already looked at some examples of predicted seeing in the form of the programs written by Roberts and Weir. We have also considered a simple example of exploratory seeing, in the form of the program written by Winston which builds models of objects from examples. The challenge for the future is to combine these approaches within a single computational model.

APPENDIX

APPENDIX

TABLE 1

(The top table shows the intensity values for a small section of the image PLANT ; the lower table gives the values of edge-mask convolutions over the same region. Only residual decay from the edge above this region is measurable. No general-purpose edge-finder could discern the edge of the nearer leaf in this part of the image.)

X = / Y	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
58	171	169	167	167	166	165	166	164	167	171	171	174	174	175	173	171
57	168	168	168	167	166	167	167	165	169	168	174	176	175	175	175	172
56	168	167	167	165	166	166	167	167	168	170	178	177	176	174	174	173
55	168	168	165	169	167	168	167	165	168	175	177	177	175	175	172	171
54	169	170	167	169	169	168	163	166	172	169	174	173	175	178	173	173
53	171	169	170	168	169	168	169	168	168	170	175	173	175	177	178	176
52	172	171	170	168	169	169	167	168	173	172	173	177	174	175	178	176
51	172	174	171	170	166	168	167	168	172	172	172	177	179	172	175	175
50	171	167	170	169	170	169	168	169	171	172	174	174	173	173	174	178
49	174	172	173	173	173	174	171	171	172	174	172	172	172	169	173	173
48	173	173	173	176	178	172	171	174	174	173	176	176	175	173	173	171
47	173	175	178	173	173	171	171	175	175	177	178	176	174	173	175	178
46	178	175	174	169	173	176	177	175	177	177	174	176	176	177	177	174
45	173	175	173	174	172	173	174	175	174	171	173	174	175	174	172	171
44	177	174	176	176	172	171	172	176	172	173	172	172	173	170	170	176
43	173	171	174	168	176	172	173	173	173	174	171	174	175	173	174	174
42	175	173	171	172	170	171	176	176	178	172	174	175	175	175	175	172
41	181	179	177	172	170	170	169	179	176	174	175	174	172	176	174	175
40	188	184	179	178	176	176	176	174	172	178	172	174	173	172	174	173
39	195	191	188	186	185	183	180	177	178	175	174	176	175	174	176	176
38	200	199	197	193	190	187	185	180	176	176	180	177	175	175	176	177
37	202	202	199	202	199	194	187	180	176	179	177	176	174	175	176	173

X = / Y	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
58	−2	−2	−2	−2	0	0	0	0	1	2	2	1	0	0	0	0
57	−2	0	0	0	0	0	0	0	1	2	2	1	0	0	0	0
56	0	0	0	0	0	0	0	1	2	2	1	0	0	0	0	0
55	0	0	0	0	0	0	0	1	2	2	1	0	0	0	0	0
54	0	0	0	0	0	0	0	1	2	1	1	0	0	0	0	0
53	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
52	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
51	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
50	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	−2	−2	0	0	0	0	0	0	1
40	0	0	0	0	1	0	0	−3	−2	−2	0	0	0	0	0	1
39	0	0	0	0	0	0	−2	−3	−3	0	0	0	0	0	1	1
38	0	0	0	0	0	−2	−3	−4	−3	−2	0	0	0	0	1	1
37	0	0	0	0	0	−3	−4	−4	−3	−2	0	0	0	0	1	1