21

Relational Descriptions in Picture Processing

H. G. Barrow and R. J. Popplestone
Department of Machine Intelligence and Perception
University of Edinburgh

Abstract

In this paper we describe work on the recognition by computer of objects viewed by a TV camera. We have written a program which will recognize a range of objects including a cup, a wedge, a hammer, a pencil, and a pair of spectacles.

A visual image, represented by a 64 × 64 array of light levels, is first partitioned into connected regions. These regions are chosen to have well-defined edges.

Having chosen the regions, the program then computes properties of and relations between regions. Properties include shape as defined by Fourier analysis of the s–ψ equation of the bounding curve. A typical relation between regions is the degree of adjacency.

Finally, the program matches the actual relational structure of the regions of the picture with ideal relational structures representing various objects, using a heuristic search procedure, and selects that object whose relational structure best matches the actual picture.

INTRODUCTION

In November 1969, a Mark I robot device (Barrow and Salter 1970) was connected on-line to the ICL 4130 computer of the Department of Machine Intelligence and Perception, University of Edinburgh.

The primary sensor of the device is a TV camera, and the computer may sample the picture at 4096 points in a 64 × 64 array, and read the picture brightness as one of 16 levels.

The device is available under the Multi-POP time-sharing implementation of the POP-2 language (Burstall, Collins, and Popplestone 1971). The program library contains functions for operating the device: for example, the function call PICINT(X, Y) returns the brightness level of picture point (X, Y).

History

In the field of object recognition, there seems to have been much preoccupation with plane-surfaced objects, presumably because they project onto a retina in a well-defined and simple manner; internal representations of these solids are easily constructed; and it is easy to deduce structure of the solid from a picture. For these reasons three 'robot projects' in the United States, at Stanford Research Institute, Stanford University, and MIT, presently restrict the environment of their devices to that of cubes, wedges, and the like.

Pictures of plane-surfaced objects are usually interpreted by fitting straight lines to edges in the picture, and then identifying parallelograms and triangles as faces of solids, and hence the solids themselves.

Guzman's program SEE (Guzman 1968), which decomposes a line drawing of a scene into sets of enclosed areas, each set corresponding to a single body, performs extremely well, and produces an analysis which is remarkably similar to that of a human observer. It depends, however, upon the assumption that all objects in view are plane surfaced.

Roberts' program (Roberts 1965), which recognizes plane-surfaced objects, does so by having an internal 3-D model of an object, computing projections from it, and manipulating them until a fit is obtained with the picture. Extension to irregular objects is by synthesizing the model from a large number of simple ones.

We had direct experience of the problems of the line finding and fitting. Murphy (1969) had investigated application of heuristic search to picture interpretation, to economize on the amount of computation required. His program did not process the entire picture in pseudo-parallel, but was guided to look at parts of it on the basis of evidence gathered so far. In this way he could find the lines of a cube, only requiring to sample 10 per cent of the available picture points.

Working with Dr R. M. Burstall, Rastall (1969) took a technique of Unger, which determined whether two graphs were isomorphic, and extended it to determining monomorphism of two families of graphs, that is, finding whether one family was a set of corresponding subgraphs of the other. Burstall suggested this might be applied to picture interpretation. A line drawing can be described by relations between the lines, such as MEET, PARALLEL, and so on. Each relation defines a graph whose nodes are the lines of the picture; an arc between two nodes means that that relation holds between the corresponding two lines. The set of relation graphs describes the picture, and similarly we may describe pictures of single objects. Thus, we can check for the existence of a given object by trying to find the object graph-family as subgraphs of the picture graph-family, using Rastall's program. Rastall himself tried this and was indeed able to find objects in pictures.

It seemed that a working object recognition program could be constructed

by combining the programs of Murphy and of Rastall, but it would not be able to handle irregular objects.

At this time we learned of the work of Brice and Fennema (1970) at SRI on region analysis. In this, the fundamental components of the analysis are areas and not lines, enabling information which is somewhat more global to be used in the processing. The analysis essentially finds the major areas of the picture.

It appeared that a description of the picture in terms of properties of regions (for example, CIRCULAR) and relations between them (for example, ADJACENT) would be better input for Rastall's program. There would be fewer regions than lines, but a richer vocabulary of properties and relations.

OUTLINE OF THE PROGRAM

The processing of a picture proceeds as follows:

(1) The picture is first completely digitized and stored in the computer core store as an array of 64×64 elements, each of 4 bits of brightness information. All succeeding processing is performed upon the stored picture, because successive samples at a point in the picture may not yield the same values of brightness due to noise in the camera and sampler, or the scene might change while processing is in progress.

(2) The picture is then analyzed into important regions, in two stages: (a) first, the picture is divided into many small elementary regions of approximately uniform brightness; (b) the elementary regions are then merged together, following a given heuristic, to produce a smaller set of larger, and hopefully significant, regions.

(3) The set of regions is then described in terms of properties of and relations between the regions (that is, as a coloured graph). The purpose of this is to abstract and generalize over a number of pictures, to sift out the information relevant to identification of objects and dispose of the rest. The properties describe shapes of regions, the relations describe their spatial and topological relationships (for example, ADJACENT, ABOVE, and so on).

(4) The description is then matched against a set of stored descriptions of views of objects. The best match (not necessarily perfect) identifies the object, by setting up a correspondence between regions of the picture and regions of the view of the object. The Unger/Rastall graph-matching technique was found to be inappropriate here, and so a different method, based upon a combination of the Graph Traverser, and Branch-and-Bound techniques was devised.

It will be noted that we are storing a set of models corresponding to objects, and we find the model which best accounts for the picture. However, the models in this case are of the sensory input, and not of the object itself (as in Roberts' case). Once the identification has been made from the picture, we may then retrieve information from a data bank concerning the object, and this may include its three-dimensional structure.

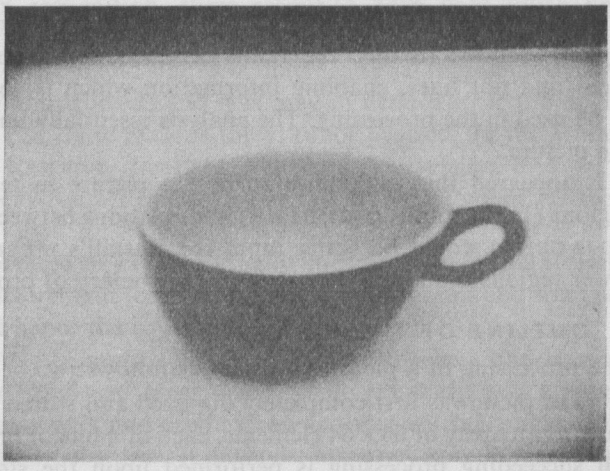

Figure 1. Teacup as seen by the TV camera, displayed on a monitor.

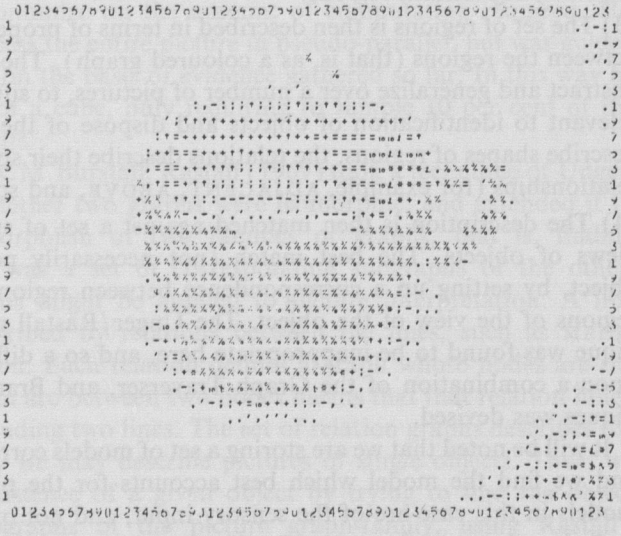

Figure 2. Digitized 'retinal image' of the input from the TV camera. The 16 discriminable brightness levels are represented by different line-printer characters.

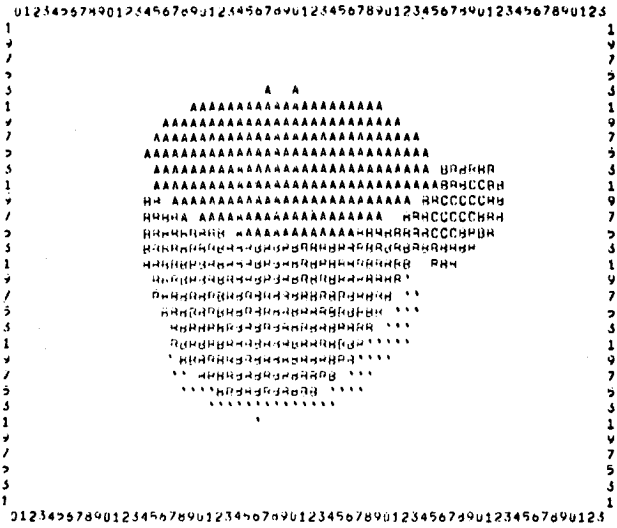

Figure 3. Region analysis of the retinal image into significant regions. Note the hole in the handle, represented by region 'c' and the shadow, represented by the region marked with the symbol '`'.

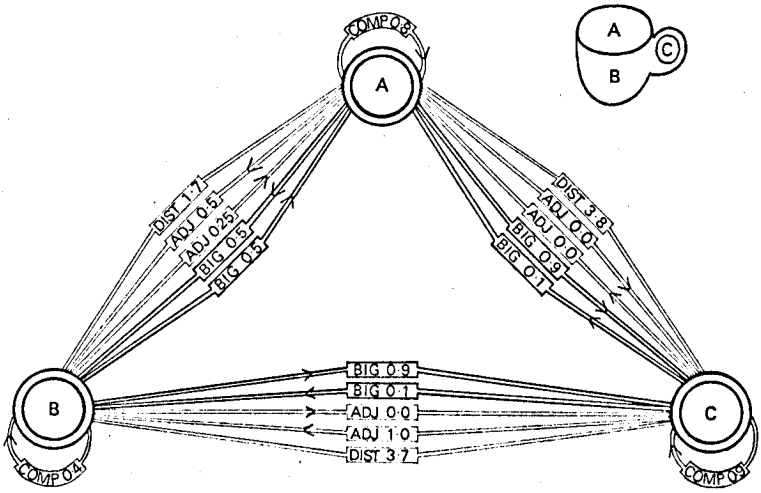

Figure 4. Computer-synthesized description of the regions in terms of property and relational measures. The numbers associated with the arcs are the measures, the names are the names of the relations. COMP ('compactness') is a shape property, and is 4π times the area divided by the square of the perimeter. ADJ ('adjacency') is the proportion of the boundary of the first region which is also a boundary of the second. Not all the properties and relations described in the text are shown in this figure.

The program will now be described in more detail.

DESCRIPTION OF THE PROGRAM

Region Finding

The first stage of the process is to find the 'important' regions of the picture. A *region* is represented by a POP-2 record which includes a description of its boundary in terms of elementary vectors, the position of its centre of area, its area and perimeter, and a membership function. This function, when applied to any point of the picture, will yield a truthvalue, which indicates whether the point lies within the region.

The simply-connectedness may seem at first sight to give rise to difficulties – what happens if a region has a 'hole' in it? In this case the hole is also a region, but one which lies within the boundary of the larger region. Points of the inner region are also members of the outer region. The smallest (minimum area) region to which a point belongs is of particular interest; it is what one would intuitively call *the* region of that point. If one wished to deal with multiply-connected regions, it would be an easy matter to describe them in terms of the simply-connected ones.

The process of finding the regions is itself composed of two phases. The first is finding a number of elementary regions of the picture, the second is merging together regions which satisfy some criterion, until no further merge is permissible and we are left with a small number of significant regions.

Brice and Fennema (1970) use an algorithm which partitions the picture completely into elementary regions, the definition of an elementary region being a connected set of points, which all have the same brightness level. This necessarily finds a large number of regions. We tried this algorithm upon a picture of a cup, with 64×64 points and 16 levels. The number of regions found was 220. Clearly this process yields a lot of data for further processing.

Our technique is not complete but much faster and more economical. The definition of an elementary region is relaxed to include points within a small range of brightness levels. We have found a suitable range to be 3 levels. Note that this relaxation makes the elementary regions larger, there are correspondingly fewer of them, but they may now overlap.

Instead of finding an elementary region for each point on the retina, we select a subset of the retinal points and find the elementary regions containing these. In our case 256 points spread over the picture in a 16×16 array are sufficient to find significant regions in most cases.

The mechanics of the region finding are as follows. Select the next starting point and determine its brightness. Step point by point towards the nearest edge of the picture until either the edge is encountered, or a point is found which has brightness outside the range of ± 1 from the starting value. If the edge is encountered, the region we are trying to find extends off the picture.

We are currently interested only in objects fully within the field of view, so the region must be part of the background. We therefore abandon this region and proceed to the next starting point.

If a point not in the region is encountered, we are at the boundary of the region. Remember this point, turn left and follow the boundary. The boundary will either go off the edge of the picture, in which case we give up and proceed to the next starting point, or will return to the first boundary point. At this point it should be stated that the boundary of a region passes *between* the picture points. Each picture point is thus surrounded by four possible elementary boundary vectors. During the walk round the boundary, note is kept of the directions of the individual steps, so a record can be constructed of the boundary curve in terms of the elementary vectors. We step round in such a direction that the region enclosing the starting point lies on the left.

Having found the boundary curve, a number of properties may be computed for the region by simple numerical integration of various functions round the curve. In this manner the area, coordinates of the centroid, and average brightness difference (contrast) across the boundary are calculated. If the curve has been followed in an anti-clockwise direction, it will bound the region externally and enclose the starting point, and its area will be calculated to be positive. If the curve closes in a clockwise direction, then we have been following an internal boundary round a 'hole': it does not enclose the starting point, and its area will be found to be negative. Regions with negative area are rejected. If they are of significant size they will be found from starting points within.

When a closed curve with positive area has been found, a region record is constructed, containing the curve and calculated properties. A check is made to see whether this region has already been noted, by comparing the calculated properties with those of known regions. Two region records with the same centroid, area, perimeter, and contrast very probably refer to the same region. If the region is unknown, it is added to the list of known regions, and a membership function is constructed for it. The membership function carries an array of one-bit components, just big enough to enclose the region: each component says whether or not the corresponding point lies inside the region. When presented with the coordinates of a picture point, the membership function first checks to see if it lies within the array, and if so it looks it up in the array.

Region Merging

When the first phase has been completed a list of region records has been constructed. The number of elementary regions found varies with the quality and content of the picture, and may be from one or two up to perhaps fifty. On average about twenty elementary regions are found.

The next phase of the region analysis is piecing together the elementary regions to yield a small number of larger, more significant regions. Brice and

Fennema (1970) use two heuristics, applied successively, the aim of which is to produce convex regions with strong boundaries.

Our present program uses the simple heuristic: merge two adjacent regions if the average contrast across the common boundary is less than some threshold (about 2.1 levels was found to give good results). Because regions overlap the notion of common boundary is slightly generalized to be that part of the boundary of one region which has a point of the other region adjacent to it and outside it.

A simple continuous shading of a surface will give rise to a series of steps in brightness level of only one unit when the picture is digitized. Thus when elementary regions are found there will be a number of them overlapping each other to correspond to the surface. The merging process will merge them all together because of the low contrast (on average, only one unit) across common boundaries. We should thus be left with a single region for this surface.

On the other hand, at an edge there will usually be a step change in brightness of two or more units so the adjacent regions will not be merged. If such a large change is not present, an edge is not distinguishable from a brightness contour by a low-level process. It requires judgements of context from a higher level, particularly if there is *no* brightness level change at an edge, as may occasionally happen.

In the program, the merging process is carried out by taking an available region and testing it against all the others to see which satisfy the criterion. Those which may be merged with it are merged simultaneously and a new region record is created. All the merged regions are deleted from the region list and the new one added if it has not already been found. This process is iterated until no further merges are possible. When this state is achieved, the average contrast across each common boundary is greater than two brightness levels.

The net result of the region analysis is the division of the picture into regions, each region hopefully corresponding to some surface of the object, and each boundary to some edge.

Finally a weeding-out procedure is entered. This discards regions which are very small (only a few picture points) and hence probably spurious, and regions which have weak boundaries, and are therefore probably part of the background (which is not explicitly represented and has already been partially discarded because it extends off the picture).

Making Descriptions

Having found the important regions of the picture, the next stage of the process is to describe the picture in terms of properties of and relations between the regions. The purpose of the describing process is to generalize. In particular it should generalize over translations, scale changes, and rotations.

The picture description contains as subgraphs descriptions of objects contained within the field of view. Any useful set of relations must allow such a subgraph to be independent of the rest of the picture; the descriptors must not be too global in nature.

Each description subgraph which corresponds to a view of an object can be so formed that it is invariant for a limited range and class of movements of the object in the field of view. (For each object, however, there may be several quite different views.) The task of the final phase of this program is to find the subgraphs of the picture description which correspond to views of objects, and hence to identify the objects themselves.

What form are the relations and properties to take? Predicates with Boolean results are obvious candidates: they would be suitable for manipulation by resolution theorem-provers. They have the disadvantage of saying very little. For example, to say that one region is bigger than another does not say whether by 0.1 or 95 per cent. One could elaborate by defining a range of predicates, each of which corresponds to a range of values of relative size. This leads to rather verbose descriptions, and extensive time and space requirements.

We use numerical measures computed from the picture for each property and relation.

Brightness of a region depends upon lighting, colour of surface, and orientation, and is therefore not a good basic measure. It may be necessary for distinguishing between black and white cats, but this is of secondary importance. *Texture* of the region might be more valuable, but is not implemented in the present program.

Shape is much more useful. It must be remembered that the shape of an image on the retina can vary considerably as the object moves and turns. However, if we restrict ourselves to identifying views of objects (that is, several descriptions correspond to a single object), then each view can be defined such that shape varies only slightly over the pictures which correspond to that view.

The properties of regions which are calculated are:

COMPACTNESS. This is 4π . Area/Perimeter^2. This measure varies from 1 (circular) to 0 (very uncircular).

SHAPEn. There are six shape components. These are derived from a Fourier analysis of the s–ψ equation of the region boundary. (*See* Appendix.)

The following relations are calculated:

BIGGER. This is a measure of relative size of regions and is Area of A/(Area of A+Area of B), varying between 0 and 1. Because it is a relative measure it is independent of scale.

ADJACENT. A measure which reflects the topology of the picture. It is the fraction of the boundary of region A which has a point of region B adjacent, or within one point of it. (The latitude has been introduced because very often a few isolated points on the common boundary of two surfaces give

rise to insignificant regions, thus interposing a gap between the two major regions.)

It will be noted that this measure is not symmetrical. In particular, if region A is inside region B then ADJACENT(A, B)=1, because all points just outside the boundary of A lie in B, but ADJACENT(B, A)=0, because all points just outside the boundary of B do not lie in A. The zero or non-zero information contained in the measure can thus provide a topological description of the picture, and the numerical value provides extra information.

DISTANCE. This measures provides geometrical information about the relative positions of the regions. It is defined as the distance between the centres of area of the two regions involved divided by the geometric mean of the average radius of the regions. The distance so calculated will be seen to be independent of scale rotation, translation and reflection. The average radius of a region is defined as 2 . Area/Perimeter.

CONVEX. This is calculated by fitting an arc of a circle to the common boundary of A and B (actually fitting a straight line to the $\psi(s)$ curve). The number calculated is the curvature of the common boundary relative to that of the circle fitting the whole boundary. (So the measure is not symmetric or antisymmetric.) That is, a result of 1 indicates the boundary to be convex relative to A and to possess the same curvature as the whole of the boundary. A result greater than 1 indicates greater curvature. A result of 0 indicates that a straight line is the best fit. A negative result means that the boundary is concave with respect to A. Apart from being just another relation between regions which helps to specify the picture, CONVEX can provide depth information in a limited sense. If the surface corresponding to one region A occludes another corresponding to region B, then it is highly likely that the common boundary of A and B will be convex with respect to A, or at least not concave. Some of Guzman's heuristics for decomposing a scene into bodies can be interpreted as using convexity of boundaries to provide statistical information about depth.

In addition to the above relations, which are adequate to describe the regions of the picture while retaining independence of scale, translation, rotation, and reflection, there are a few further relation measures:

ABOVE. This measure is similar to DISTANCE and is the vertical distance between the centroids of the regions involved, normalized by dividing by the average radius. If positive, A is above B, if negative, B is above A. The inclusion of ABOVE immediately removes independence of rotation in the plane of the picture. Objects are usually encountered in a preferred orientation, so inclusion will aid identification in the normal case. (As will be explained later, if objects are presented during training in many orientations equally often, the weight attached to this measure will be reduced, so it does not hurt to include it.)

BESIDE. This is similar to ABOVE. It is the horizontal distance between centroids, normalized. However, the sign of the result is always positive, so

that BESIDE(A, B)=BESIDE(B, A) and the measure is independent of reflection. The inclusion of this relation again reduces the independence of rotation, but assists when objects are usually seen in a particular orientation. The reflection independence has been retained because there are often two views of an object which are almost mirror images, for example, left and right profiles of a face, a cup with handle on left or on right. Whereas, up-down symmetry is rare. Turning an object upside down does not simply invert the picture unless it is viewed exactly from the side.

The set of properties and relations above seems to be useful and powerful. It could be extended greatly, but at the expense of processing space and time requirements.

The description is generated exhaustively for experimental convenience. A more practical version of the program would only compute the properties and measures it required, when they were required. [Using the memo-function idea – *see* Michie (1968).] From the point of view of the next stage of the processing it would nevertheless appear that a complete description was available.

THE DESCRIPTION-MATCHING PROCESS

Having produced a description of the picture, the next stage of the process is to interpret it. For the purposes of the present research, certain assumptions were made. The picture is assumed to contain a view of a single object, which is wholly contained within the field of view. The aim of the process is to decide which of a predetermined set of views of objects most resembles the picture, and hence to identify the object in the picture.

The picture will contain a number of regions which are not part of the target object. The representation of the object in the picture may be degraded from the ideal by the addition or deletion of regions: for example, a highlight on a surface may appear as an extra region, a hole in a surface may not be detected.

There are a number of situations with which the existing program cannot cope adequately. If a surface has a hard shadow across it, it may result in two regions. An obvious method of overcoming this situation is to merge the offending two regions into one and carry on. Since many such merges are possible, this approach has not yet been investigated.

It is possible for two surfaces meeting at an edge to be lit so that the edge is practically invisible. The region analysis will find one region instead of two. A simple way of coping with this is to store the descriptions of likely degradations of views among the target set. Again this is not very economical, and has not been investigated.

The effects of occlusion of objects are varied. If the occlusion is slight, performance is not affected. As the occlusion increases, the property and relation measures will change. The matching process can still function correctly when a match is imperfect and so may be able to identify the object correctly.

Strategy

The description of the picture being analyzed, and the descriptions of views of objects are similarly represented. For each, a record of two components exists. One component is in essence a list of formal parameters. The regions of the picture are represented by numbers 1, 2, 3, etc., those of a view by letters A, B, C . . . . The second component is a list of relations (properties of regions are treated as two-argument relations, of which the second argument is identical to the first).

For each view there are many ways in which the regions of the picture may be put into correspondence with those of the view. We can define a correspondence to be *complete* if for every region of the view a corresponding region of the picture has been assigned. Otherwise a correspondence can be said to be *partial.*

For a relation, REL(A, B) of the view, if the picture regions corresponding to A and B are defined, we can find the corresponding picture relation. The view relations have mean and standard deviation stored. The corresponding picture relation is said to *agree* with the view relation if its measured value lies within three standard deviations of the mean. Thus for any correspondence between region sets, it is possible to determine how many picture relations can be tested against view relations, how many of these agree and how many do not (known as Tries, Successes, and Failures). It is then possible to evaluate a particular correspondence and assign a score to it indicating its merit. The aim of this stage of the program is to find the most valuable correspondence of all possible correspondences, both complete and partial. Note that if the region-finding process has met with difficulties and has lost a significant region or split it into two, it may be better to accept a partial correspondence rather than force an incorrect assignment of a picture region to a view region.

The problem can be restated as finding the best match of a subset of the picture regions with a subset of the view regions. Unger's graph isomorphism finding procedure and Rastall's generalizations of it are not adequate. Rastall's procedure handles only the matching of a complete graph with a subset of another.

Evaluation function

Before continuing, it is worthwhile to consider the choice of a suitable function for evaluating correspondences.

The number of relations for a 1-region object description is 7, for 2 regions 28, for 3 regions 63. Suppose we take the number of successes to be the value of a correspondence. It is clear that a perfect match with a 1-region view will be rejected in favour of a bad match with a 3-region object, which is unsatisfactory. The situation is better if fractional success is used to evaluate. We have introduced a measure of the degree of failure. If computed picture

measure lies within 3*sd* of the mean, the failure is 0, if between 3 and 6*sd*, failure is 1, between 6 and 9, failure is 2, and so on.

Tries − Fails is thus a more sensitive measure than Successes.

There is still a dilemma apparent. At what point does one reject a good match with a simple object for a not-so-good match with a complex object? Intuitively one feels that a match with a complex object is preferable to an equally successful one with a simpler object. It is unfortunately rather easy to find simple objects as parts of complex ones.

The evaluation function chosen works reasonably well, though there is room for improvement. The program endeavours to minimize the function, so it has the following form:

$$1-\frac{\text{Tries}-\text{Fails}}{\text{No. of Relations}}+\frac{K}{\text{No. of Regions}}$$

where No. of Relations and No. of Regions refer to the view description involved. A suitable value for K appears to be 0.5.

Tactics

A rather naive method for finding the best match would be to generate and evaluate all possible correspondences. We can, however, improve on this.

Beginning with a partial correspondence, say with n assignments of picture regions to view regions, it is a straightforward matter to generate a correspondence of $(n+1)$ assignments from it, by choosing an unassigned picture region and pairing it with an unassigned view region. In general, a set of correspondences with $(n+1)$ assignments may be so derived. Let us define this process to be the *development* of a correspondence. The development can proceed step by step if we generate new correspondences one at a time. A correspondence which has not yet had all immediate successors developed from it is defined to be *partially developed.*

To evaluate a new correspondence we can use the information computed about its parent, and need consider only the consequences of adding the extra pair of regions.

At the start of the process we have only the null correspondence. From this we can generate all first-order correspondences, those with only one pair of regions, and from those, the second-order correspondences, and so on. It makes sense now to develop only the most promising correspondence, and further to generate only one successor at a time.

So far the process is analogous to that of the Graph Traverser (Doran and Michie, 1966; Marsh, 1970). At a given instant there are a number of jobs to be done, each corresponding to a partially-developed correspondence, and all at various stages and degrees. By working only on the best available job we can economize in effort. There is a difference between this and classical graph traversing – in this case we do not necessarily know the goal state when we encounter it, because we are looking for the best.

At this point, it appears that developing the best available correspondence was an illusory advantage, since it seems we must search the whole space anyway. Not so. Given a correspondence, not only can we evaluate it, we can also determine upper and lower bounds for the values of all its successors, because no successor can have fewer successes or fewer failures than it has. Thus we can incorporate 'don't develop a correspondence if its successors cannot give better values than the most successful so far found'. (This uses the same pruning technique as the Branch-and-Bound method – *see* Burstall 1967.) We need to remember only the best correspondence so far encountered, and at the end of the process, when there are no more promising lines of development, this will be the answer, the best match.

This hybridization of the Graph Traverser and the Branch-and-Bound algorithm is complete because it cannot fail to find the best node in the search space, but is more efficient than an exhaustive search procedure. In this particular case, the final result, the best match, may be a partial or a complete correspondence. In practice, we also retain matches which are almost as good as the last.

TEACHING THE PROGRAM TO RECOGNIZE

The facility was provided in the program for enabling it to 'learn' in the light of experience and guidance. It was not originally intended to exist, but when the nature of relations and properties became established, it was realized that such a facility could easily be provided and would ease considerably the problems of constructing models.

It will be recalled that the relations in a model are in fact measures, and the best way to obtain values for the model is to make measurements on several pictures of the object and to calculate the mean. If the difference between the measure and the mean is used to compute badness of match, then the identification made by the program will be unduly swayed by those measures which are least reliable, that is, those which have the most variation. As we are calculating means, it is a simple matter to calculate standard deviations as well and to replace the discrepancy measure by deviation in units of standard deviation. Compensation is thus introduced automatically, so that all measures carry the same weight. The probability of exceeding $3sd$ is roughly the same for all the measures.

It will be noted that, since we are using numerical measures, the adaptation mechanism which is appropriate, namely, simple statistical calculation, is well understood. If the measures were binary, a system of weighting would probably have to be introduced.

The training procedure is as follows. First, the object is placed before the TV camera and the picture is analyzed into regions. The regions are exhaustively described in terms of properties and relations. (The matching phase of the program may be entered, if desired, to see what the program would guess the object to be.) The command 'LEARN' is then given, together with

two vital pieces of information. The first is the object (or view of an object) which is the correct response, and the second a correspondence, which explains which regions in the picture correspond to which in the view. Comparisons between picture and view descriptions may be made before and after the updating process, to see what discrepancies existed and whether they remain. Usually they are eliminated by the learning process.

The provision of the correspondence may appear artificial. The learning process was not intended to be used sportingly, but rather to save time in supplying the model data. With a little modification, however, it could be made more flexible. For example, instead of supplying the correspondence, the program could be made to replace the list of views of objects against which it matches the picture by a list of only one, the specified object (or perhaps by the views of that object). The matching process could then be entered, and the best correspondence found and used in the updating process. Occasionally the correct correspondence might not be found, but in such cases the picture would have to be ambiguous, or degraded. Perhaps the updating process could be inhibited if the discrepancy is too great, indicating possible gross error.

DISCUSSION AND CONCLUSIONS

First, the program can recognize a variety of objects, both regular and irregular, when presented singly, in standard positions and diffuse lighting. It can distinguish on the basis of shape (ball *versus* pencil) as well as complexity (tube *versus* cylinder) and region relationships (cup, spectacles). It has nine objects in its current repertoire, and there is no doubt that this could be considerably extended.

It is difficult to assess performance, because it depends upon the training given. If a cup is presented in nearly the same position every time, there will be little tolerance in the learned description. Since identification proceeds on the basis of the *best* match, cups in non-standard positions may still be correctly identified, but may also be misidentified if other objects of the repertoire exist with sufficient tolerance. If the cup is presented with the handle hidden behind, whereas all learning had been made with the handle nicely out to the side, it would probably be misidentified. If the program were forced to learn this view as simply a form of the one view of the model, the consequence would undoubtedly be bad and subsequent performance degraded. At present the onus is upon the operator to decide when a new view should be created, though conceivably a program could be written which would do so if the match with existing views was sufficiently bad. The performance of the current program depends upon the training process. If a new picture can be described correctly by a model description, that model will inevitably be returned as a possible interpretation of the picture. If not, then the best fitting model will be chosen.

RESULTS

With the above reservations, some experimental results are recorded in the table. Ten test objects were used to teach the program nine categories (there were two cups of different sorts). Each object was presented during teaching at least six times.

Table 1. Performance of Object Recognition program: three trials with each of ten objects (including two cups). [Percentage correct: 85. Average time: 5 min 40 sec.]

	IDENTIFICATION								
OBJECTS	*Pencil*	*Ball*	*Hammer*	*Wedge*	*Cylinder*	*Doughnut*	*Tube*	*Cup*	*Spectacles*
Pencil	3								
Ball		3							
Hammer	2.5		0.5						
Wedge				3					
Cylinder					3				
Doughnut						3			
Tube					1		2		
Cup								6	
Spectacles				1					2

An assistant, unassociated with its training, placed objects in the field of view of the camera in accordance with brief written instructions (for example, CUP: hole in handle must be visible, right way up). Each object was presented 3 times, in different positions.

As can be seen from table 1, objects were correctly identified 25.5 times out of 30. (The 0.5 result was a dead heat between hammer and pencil.) The rate of success is about 85 per cent.

Limitations

There are two obvious limitations: time and space. The program takes about 5 minutes 40 seconds on average to analyze a picture (though the time may range from 70 secs to 15 mins). It must be remembered that it is written in a high-level language and could probably be speeded up by a factor of 10 by machine coding.

Program, stored picture, region data structures, picture and object descriptions, together occupy about 20*K* of 24 bit words. Some monitoring and utility functions not concerned with the mainstream of the analysis could be removed, while again machine coding could probably effect improvements.

There are some logical limitations. If pictures or objects with more than about 6 regions are involved, matching time climbs rapidly. This is due to the single-level nature of the matching process.

A further major limitation is the present set of properties and relations. They are rather too global in nature. If an object is gradually occluded, some of its regions become more and more distorted; the properties and relations are significantly affected. Because of the globality, the description is too sensitive to local changes.

Future Work

The most severe limitation upon the present program is probably the restriction to single objects, though it is by no means drastic or insurmountable. The extension to multiple, but non-occluding, objects is almost trivial; one simply has to be more discriminating about matches remembered and discarded and employ the rule that no region may be part of more than one object.

To extend the ability of the program to handle occlusion, the main requirement is to add local information to the picture descriptions. Boundaries of regions must be split into components and the shapes of these described individually [similar to the scheme proposed by Guzman (1971)]; junctions between regions could be described and related. Ideally, the description should contain information ranging continuously from very local to global, and describing regions, curves, and points with much redundancy; it is the redundancy which makes the scene interpretable even when parts are apparently missing.

Such a comprehensive picture description brings its own disadvantages, namely, a great burden is thrown upon the matching stage of the program. A single-level search would take an inordinate amount of processing time. What must be developed is a program which performs the matching hierarchically, that is, which first pieces the simplest components together into simple compounds, then pieces these together, and so on, up to the level of objects, or groups of objects. It seems that any such program would need the ability to direct its activities in the light of its discoveries, in a manner closely similar to that of Murphy. The mechanism exists in the present program for constraining the area of search in the picture. It has been used in a simulation of picking up objects. In an integrated vision program, the 'directing of attention' would proceed at many levels, with feedback from higher to lower processes.

The principles underlying the present program, namely, 'analyze, describe, match' have wider interpretation. Although we have chosen the object models of this paper to be of two-dimensional pictures, they can be of three-dimensional solids.

It is possible to find a number of picture properties and relations which provide information about the solid with a fair degree of reliability. In the case of plane-surfaced polyhedra, two adjacent regions of the picture cor-

respond to adjacent surfaces of the solid, an angle less than 180° in the picture corresponds to one of less than 180° in the solid, the number of sides a region possesses is the same as the number of sides of the corresponding surface. A little experimentation was performed with such properties and relations forming descriptions of picture and model. For example, a view of a cube (3 adjacent regions) was matched against a model of a cube (6 surfaces), and the 48 possible interpretations were obtained as results.

One interesting application of the techniques is to the interpretation of stereo pairs of pictures. The main difficulty lies in deciding which points in the two pictures correspond (simple cross-correlation is fraught with problems). If the picture is simple, the matter is relatively straightforward but, otherwise, the pictures must be interpreted and objects identified before depth information can be extracted. We propose the following: first, analyze both pictures into regions; second, describe them; third, match the two picture descriptions.

The result of the match will be the best correspondence that can be found between the two pictures; it is a list of pairs of regions, one from each picture. As we already compute centroids of regions, we can compute average range of a region from the displacement. We may now, for example, add the range information to one picture description and match against models, or use the information directly to divide the regions into groups, analogous to Guzman's decomposition.

This process has the advantage that we require no elaborate program specially for range-finding or decomposition. The same set of functions will perform this and identification too.

Lastly, let us consider the object recognition program in its proper perspective, as a part of an integrated cognitive system. One of the simplest ways that such a system might interact with the environment is simply to shift its viewpoint, to walk round an object. In this way more information may be gathered and ambiguities resolved. A further, more rewarding operation is to prod the object, thus measuring its range, detecting holes and concavities. Such activities involve planning, inductive generalization, and, indeed, most of the capacities required by an intelligent machine. To develop a truly integrated visual system thus becomes almost co-extensive with the goal of producing an integrated cognitive system.

Acknowledgement

The authors would like to acknowledge the financial support of the Science Research Council throughout the work of this paper, and also a research contract from the GPO Telecommunications Headquarters in aid of the robot project.

REFERENCES

Barrow, H.G. & Salter, S.H. (1970) Design of low-cost equipment for cognitive robot research. *Machine Intelligence 5*, pp. 555–66 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Brice, C.R. & Fennema, C.L. (1970) Scene analysis using regions. *J. Art. Int.*, **1**, 205–26.
Burstall, R.M. (1967) Tree-searching methods with an application to a network design problem. *Machine Intelligence 1*, pp. 65–87 (eds Collins, N.L. & Michie, D.). Edinburgh: Edinburgh University Press.
Burstall, R.M., Collins, J.S. & Popplestone, R.J. (1971) *Programming in POP-2.* Edinburgh: Edinburgh University Press.
Doran, J.E. & Michie, D. (1966) Experiments with the Graph Traverser program. *Proc. R. Soc. A*, **294**, 235–59.
Guzman, A. (1968) Decomposition of a visual scene into three-dimensional bodies. *Proc. Fall Joint Computer Conference*, pp. 291–304. Washington, DC: Thompson Book Company.
Guzman, A. (1971) Analysis of curved line drawings using context and global information. *Machine Intelligence 6*, pp. 325–75 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.
Marsh, D. (1970) LIB GRAPH TRAVERSER, *Multi-POP Program Library.* Edinburgh: Department of Machine Intelligence and Perception. Reproduced in Burstall, Collins & Popplestone (1971).
Michie, D. (1968) 'Memo' functions and machine learning. *Nature*, **218**, 19–22.
Murphy, A. (1969) An application of heuristic search procedures to picture interpretation. *Research Memorandum MIP-R-61.* Edinburgh: Department of Machine Intelligence and Perception.
Rastall, J.S. (1969) Graph family matching. *Research Memorandum MIP-R-62.* Edinburgh: Department of Machine Intelligence and Perception.
Roberts, L.G. (1965) Machine perception of three-dimensional solids. *Optical and electro-optical information processing*, pp. 159–97. Cambridge, Mass.: MIT Press.
Rutovitz, D. (1970) Centromere finding: Some shape descriptors for small chromosome outlines. *Machine Intelligence 5*, pp. 435–62 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

APPENDIX

Rutovitz (1970) has experimented with describing the shapes of chromosomes, represented as closed curve outlines in polar coordinates by a Fourier analysis of r as a function of θ. He takes the origin of the coordinate system to be at the centroid of the enclosed area, but this is unsatisfactory because of the dependence of the results upon the choice of origin. Moreover, the technique is restricted to curves which are single valued in r for given θ.

Any curve may be represented in s–ψ coordinates, where ψ is the angle the tangent to the curve makes with a given line (usually the X-axis) and s is the length along the curve between the point under consideration and an arbitrary zero point. The function $\psi(s)$ is single valued, making it suitably unambiguous for practical curves of arbitrary complexity.

If a closed curve is considered, as we increase s, ψ increases by 2π for each complete circuit round the curve. Consider now $\phi(s)=\psi(s)-2\pi \,.\, s/s_0$ where s_0 is the circumference of the curve, so we have subtracted out the steadily-rising component. The function is single valued and, moreover, repeats cyclically because $\phi(0+s)=\phi(s_0+s)$. It is therefore eminently suited to Fourier analysis. The shape properties mentioned in the paper are the r.m.s. amplitudes of the first 6 Fourier components of the function $\phi(s)$. The relative phases of the components are not used.

$\psi(s)$ and $\phi(s)$ have some interesting properties. A circle has constant $d\psi/ds = 2\pi/s_0 = 1/r$ and so $\psi(s)$ is a straight line of slope $2\pi/s_0$ and $\phi(s)$ is constant. Thus if part of the curve is an arc of a circle, it will transform to a straight line on the s–ϕ or s–ψ plots. The magnitude and direction of the slope of the line give the radius and sense of the arc. If a part of the curve is a straight line, this again transforms to a straight line on s–ψ (constant ψ) or s–ϕ plots.

Thus, fitting a straight line or an arc of a circle to a given curve, both reduce to fitting a straight line on the $\psi(s)$ graph.

Discontinuities of gradient (kinks) of the original curve transform to jump discontinuities.

The CONVEX(A, B) measure, described in the paper, is computed by fitting the best straight line to the $\psi(s)$ representation of the common part of the boundary. This gives its average curvature, which is then normalized by dividing by the average curvature of the whole boundary of region A (that is, by $2\pi/s_0$).

Comment added by Dr A. Rosenfeld

Fourier expansion of the Polar equation of a boundary curve was proposed in the 1950s (US Defence Documentation Report No. AD 148612). Dr Jerome R. Cox, Jnr, of Washington University was also working along similar lines in 1969.