13

Object Recognition using Isolumes

Rory C. Flemmer, Huub H. C. Bakker and Claire L. Flemmer
Massey University, Palmerston North
New Zealand

1. Introduction

1.1 The need for object recognition

Workers in artificial vision (AV) would say that their basic aim is to be able to take any picture and recognise the objects present and their orientations. They would say that the problem is difficult and that there are many sub-specialties and that they are working in one of them. What is the point of the endeavour? Well, it's going of be of benefit to mankind – for security or autonomous robots. But we would reply that Rottweilers are pretty good at security and are totally incompetent at recognising objects in pictures. It is therefore clear that the skill of recognising objects in pictures is not required for an agent to go about its business in the real world. Of course, CCD images seem rather like pictures when they are presented on a computer monitor but, in fact, they supply very different information from that gathered by a biological vision system operating in the real world. Visual systems in the biosphere have been around for over half a billion years (Valentine et al. 1999, Levi-Setti, 1993 and Conway-Morris, 1998) and most segment objects by motion – many animals cannot see stationary prey. In addition, the more sophisticated ones use stereopsis and information from parallax (consider the way a chicken moves its head as it examines something). It is only once we rise up the evolutionary chain to the best of the primates that the huge library of experience contained in the cortex can be used to offer a syntax of vision. If we were to rank the accomplishments of artificial vision against those of biological vision, we would have to place its current proficiency considerably below that of a cockroach or a spider – and these humble creatures function quite adequately in the world.

It is the glory of modern science and human connectivity that it is able to reduce and distribute problems to many workers and then aggregate the results of their efforts. In the case of AV this has resulted in the subspecialties having a life of their own with their own techniques, vocabulary and journals. It is not within the purview of such specialist researchers to take an integrated view of the whole endeavour and to ponder its aims and terms of reference. We, however, are striving to produce an autonomous, intelligent robot. This ineluctably requires vision capability of a high order. We are therefore forced to consider ways and means to accomplish this. Accordingly, we examined the voluminous literature and sought among the technologies of object recognition to find a working method. We sought in vain. Nobody can do a robust and adequate job of finding objects in an image.

From any elementary book on human vision (for instance, Gregory 1978), it is immediately obvious that the strategies used by biological visual systems as they deal with objects in

three space, bear little relation to the problem of examining a static and complex two dimensional image. Furthermore, the problem of examining a complex two dimensional image is very, very hard – you have to have a top-end primate brain to do it. It is not surprising therefore that sixty years of AV, aimed almost exclusively at the picture problem have produced but modest accomplishments.
But we still want to build an embodied artificial intelligence and therefore we have to have vision.

1.2 Why do existing techniques not deliver?

One of the subspecialties of vision is segmentation of the image into discrete objects. This is appropriate because researchers tacitly, perhaps even subconsciously, agree that vision is all about objects. We go a lot further and have argued (Flemmer, 2009) that life and intelligence are about nothing other than objects. Unfortunately AV is not generally able to segment objects in a picture – once again, it takes an advanced primate brain to do this. However, the spider and the cockroach, with their modest intellectual machinery, have this problem handled without breaking a sweat. We can readily imagine how they might segment by movement or parallax. Of course these techniques do not work on a two dimensional image. Given a segmented image, the task of delineating the outlines of objects obviously becomes very much simpler.
It seems therefore that, if we consider the AV problem from the point of view of an agent going about its business in three-space, with stereo cameras, it will be somewhat easier, if for no other reason than that the segmentation problem becomes tractable in most situations. However, we are still faced with the task of making sense of a segmented portion of an image. There are no workable technologies to handle this, even when it is segmented and even when we can reduce it to a fairly simple image by considering only the extent of a particular object.
Can current AV analyse a fairly simple picture? It seemed to researchers in the 1980's (for example, Rosenfeld, 1987), when they had had twenty years to explore AV, that the way forward was, firstly, to find the outline of an object (its cartoon, derived from an edge-follower, together with some help from segmentation techniques) and then to recognise the object from the shape of the cartoon. They ascribed their lack of success to the poverty of their computers compared with the human brain (Ballard and Brown, 1982). Comparisons with cockroach brains were not reported. In the intervening thirty years, computers have improved 30,000 - fold (Hutcheson, 2005) and we have had the best and the brightest minds considering the problem. But in a recent review of object recognition, Da Fontura Costa and Cesar, 2009, note that "computer vision systems created thus far have met with limited success". In particular, they observe that edge-followers do not robustly yield edges, independent of lighting conditions, noise, occlusions and distortions.
For thirty years, a benchmark for edge followers has been the image of Lenna, a 1972 Playboy centrefold, shown below (Fig. 1) in modest quantities (Hutchinson, 2001).
Nor all our piety nor wit can satisfactorily find the edges of the sunny side of Lenna (Fig. 1a), despite decades of extremely close attention by graduate students. Even less can we find the edges of her reflection in the mirror (Fig. 1b) and even if we could, we could not reasonably conclude that the resulting cartoon was that of a comely maiden, although our own examination of Fig. 1b might suggest this. It is hard to see how the paradigm of edge detection and cartoon recognition could work in this case – and it is not an extraordinarily

(a)

(b)

Fig. 1. (a) Image of Lenna (b) Reflection of Lenna

difficult image. A more compelling indictment is the sober fact that the scheme has failed to solve the problem despite fifty years of earnest endeavour and millions of person-hours. It is suggested (for example see Ballard and Brown, 1982 and Da Fontura et al, 2009) that the impasse can only be resolved by introducing a visual syntax to guide us in finding edges, i.e., we need a human cortex to help. This seems technically difficult.

Lately some progress has been made using Scale Invariant Feature Transforms (SIFTs). The notion is that objects have certain idiosyncrasies which, if they were scaled and rotated, would always show up in different images of the same object, if viewed from roughly the same perspective. This technique suffers from the intrinsic problem that a polka dot mug is not seen as similar to a tartan mug. Nonetheless, the technique has produced some very respectable object recognition (Brown and Lowe, 2007) although it cannot be regarded as an overarching solution.

Other techniques are reported in a comprehensive review (Da Fontura Costa and Cesar, 2009) but we judge that none of them springs forward, fully formed, to solve our problem and none is as important in the literature as cartoon creation followed by attempted recognition (Drew et al., 2009).

1.3 How do we go forward?

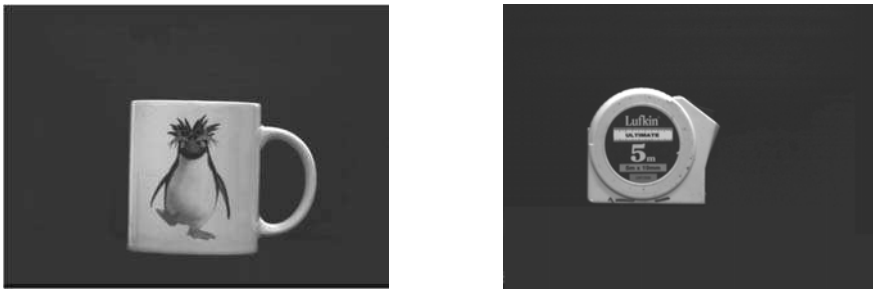
Let us accept that edge-followers, despite being a very mature technology, do not work to a level which allows robust object recognition. We view level sets as a subset of edge-followers (Osher and Fedkiw, 2003, and Sethian, 1999). Let us accept also that SIFTs and other techniques do not provide an immediate prospect of competent unsupervised AV.

In this chapter, we offer an avenue that might prove useful. This is the concept of iso-luminal contours – or isolumes. We have deployed this concept and elaborated it to be a method that has had some success in unsupervised object recognition. Despite the modest dimensions of our efforts and of our success, we hope that, given some attention from the massive intellectual resources of the AV community, this scheme might lead to robust unsupervised object recognition.

1.4 How will we know whether our scheme is satisfactory? Choosing a database

It is customary to test object recognition methods against image databases such as the Caltech-101 database (Amores et al., 2007), the COIL-100 database (Scheneider et al., 2005),

the MIT-CSAIL database (Gao et al., 2007), the ETH-80 dataset (Dhua and Cutzu, 2006), the MPEG7 database (Belongie et al., 2002) and the Corel stock photography collection (Lew et al., 2006). This has the merit that, aside from the progenitors, the authors are viewed as comparing against a fixed standard. But our requirement is not merely the 'solution' of the object recognition problem but actually to deploy a technology for the use of an artificial intelligence. What, then, is an appropriate photographic database? Clearly it is the succession of images captured by the cameras as the robot goes about its daily round. Since this is not yet available, we have created a database (which is accessible at the URL in the appendix). This database differs from others in that it caters for our specific requirements; namely that we have specific and exact images which are our gold images. We seek to judge whether they are present or not in the other database images. This is rather different from most databases which might have, for instance, a succession of cars, all slightly different.
Our database contained two gold exemplars and 100 brass images. The gold exemplars are shown in Fig. 2. Some of the 100 brass images contain gold images, some occluded, some frank, at varying orientations and scales.



(a) (b)

Fig. 2. Two gold images used for testing; (a) a mug and (b) a measuring tape.

A selection from the brass image database is shown in Fig. 3.

2. Isolumes

Imagine that a monochrome image were placed flat on the desk (Fig. 4a). Imagine that the grey level at each i-, j-position in the image were plotted as a height above the table (Fig. 4b). This means that we can view the image as a topography. We can then connect points of equal grey level to form contours as land surveyors do with points of equal altitude. We call these contours ‘isolumes’. An isolume is outlined in yellow in Fig. 4a and its corresponding representation in Fig. 4b.

2.1 Representation of objects

When we take an electronic snapshot of an object, we will see it from one definite viewpoint. But an object looks different from different perspectives, as it is rotated. To handle this, we propose recording each object as twenty views, each along the axis of a regular icosahedron (a regular polyhedron with 20 identical equilateral triangular faces). Thus each object is represented in our database by twenty 'gold' views. Such views will be 41.6 degrees apart and it is assumed that an object can be recognised provided that it is not rotated by more

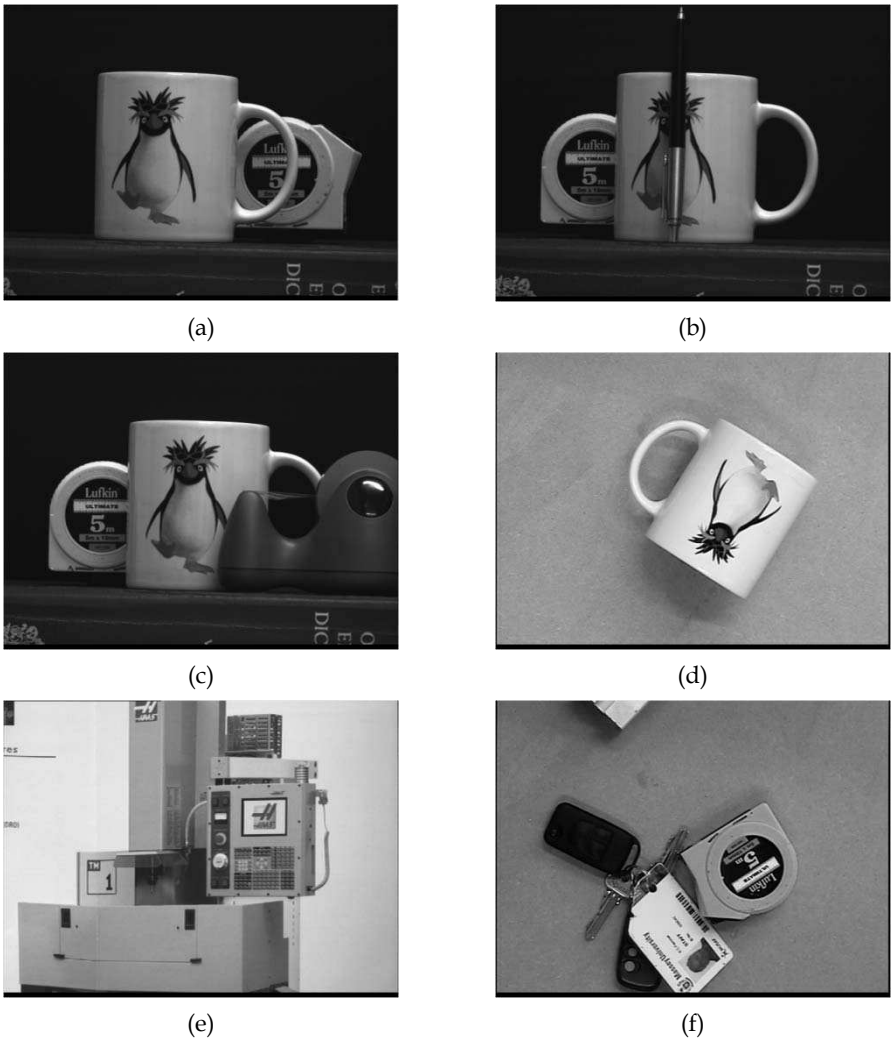
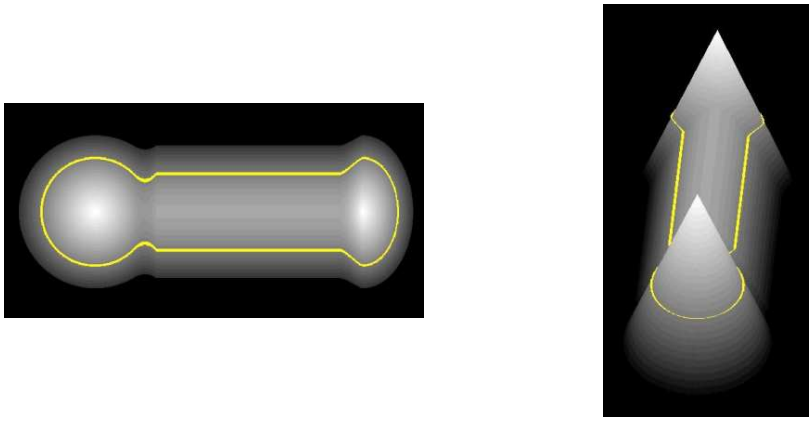


Fig. 3. (a) to (f) Six examples of images in the brass dataset

than 21 degrees away from its archival image. (We confirm this fact experimentally in due course.) With how many objects might we have to contend in our database? Educated people recognise of the order of 45,000 words (Bryson, 1990). By conducting an assay of the Oxford Dictionary, we find that about a third of these are nouns, so let us say there are 15,000 objects in our ken. An objection arises in that 'car' encompasses many models, with new ones added every year. In order to deal with this, we will need to invoke the concept of universals – which has plagued philosophers from Plato onwards. We will set this aside for the moment and consider that our library contains only specific objects and each object is recorded as twenty views. Thus 'Ford Focus 2009' and 'Ford Focus 2010' are, for the present, two distinct objects. We will return to the problem of universals later.



(a) (b)

Fig. 4. (a) A single isolume shown in yellow and (b) its height representation

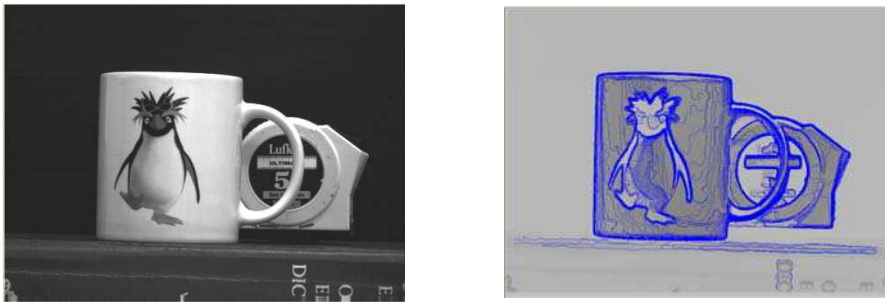
2.2 Extraction of Isolumes from an image

In principle we can extract an isolume for every grey level in the image. In practice, it is wiser to establish the range of grey levels present in the image and then acquire isolumes at some grey level increment (not necessarily constant) so that we obtain a lesser set of isolumes to represent the image. Depending on the image, we find it convenient to use perhaps forty grey levels and we will get something like five or ten isolumes for each of these. We find, experimentally that these numbers yield a good description of the image. It turns out that we have the further requirement that we have to extract the isolumes to sub-pixel accuracy; an accuracy of 0.1 pixels is barely good enough. Before analysing the image, we introduced some Gaussian blurring so that the very sharp cliffs of the topography are given a gentler slope. Consequently, the isolumes are spaced out a little and present as a broader line. Fig. 5 shows a complex image and the isolumes of the image in blue. Intuitively it seems that they capture the sense of the image very adequately.

Where many of them coincide, they present as a solid line on the image and we would consider such a line to be an 'edge' in the traditional sense of image analysis. In fact, we could select only those multiply coincident contours and consider them as edges.

Our isolume extraction process can be viewed by examining the C# code which can be downloaded from the URL in the appendix. Undoubtedly, those who follow will do it faster and better but this code is adequate for our immediate needs. Fig. 6 shows the process which has the following steps:

1. Create arrays for RoseI(24) and RoseJ(24). These arrays specify a set of vectors such that (RoseI(0),RoseJ(0)) points along the positive I axis, with length 4, i.e. RoseI(0) = 4, RoseJ(0) = 0. For the index equal to 6, the vector points along positive J. This device permits a step to be taken from a current point in any of 24 directions by setting the index.
2. Specify grey level.
3. Create a Boolean Incidence Array of the same dimensions as the image. Call it StartPoints(). Step through the image at intervals of 5 pixels in I and J. Set the element to be true if the image grey level is close enough to the specified grey level.



(a) (b)

Fig. 5. (a) Image (b) Isolumes of the image

4. Create a second such integer array to store the index value of a point on an isolume, IsolumeIndex. Call the array IsolumePoints().
5. Top of Loop
6. Search through the Incidence Array until a True element is found. Set this element False and start tracing the isolume at this set of coordinates.
7. Search in a circle around this point using the vector (RoseI(k), RoseJ(k)) for k = 0 to 23, to find that pixel which is closest to the specified grey level. In general there will be two such points. Choose between these points such that, as the isolume progresses from the initial point to the second point, bright is on the right. This rubric simplifies the isolume structure.
8. Interpolate between neighbours to this pixel to provide values of the isolume coordinates to sub-pixel accuracy (better than 0.1 pixel).
9. Use of the RoseI/RoseJ stratagem provides points on the isolume at intervals of about four pixels. Interpolate linearly to provide another point in the middle. Record each point as IsolumeX(IsolumeIndex) and IsolumeY(IsolumeIndex).
10. Set these two elements to the appropriate value of the IsolumeIndex in the IsolumePoints() array. Later, this will allow a very rapid search for the case where the isolume crosses itself.
11. Check that the isolume is not circling around and crossing itself. This can be done by scanning a 6x6 block of entries in the IsolumePoints() array, centred on the new point and demanding that any value found not be more than four points different from the IsolumeIndex, i.e. ignore the neck of the snake but be alert for its body.
12. If the current point is approaching a point which has already been seen on the snake or else is stepping out of the picture, then go to the start point and complete the isolume in the other direction.
13. Go to the top of the loop (step 5).

2.3 Manipulation of raw Isolumes

Once we have the isolumes, we elect to plot them (Fig. 7) as local curvature versus distance along the isolume and call this plot the fingerprint of the isolume. As we will see later, we fit a local circle to the isolume and use the reciprocal of its radius as the curvature. Consider Fig. 7 where the isolume defining the outer edge of a mug is plotted with these coordinates. Such a plot provides three distinct features; there are lobes, which present as bumps (and are marked with asterisks on the fingerprint), lines which present as portions of zero

curvature (marked with lines on the fingerprint) and arcs, which have constant, non-zero curvature (and are marked with an arc on the fingerprint). Observe that the area under a lobe represents the amount of rotation of the isolume and is scale and rotation invariant. Also note that the order of the features as they appear on the plot is independent of scale and rotation, although it might be reversed.

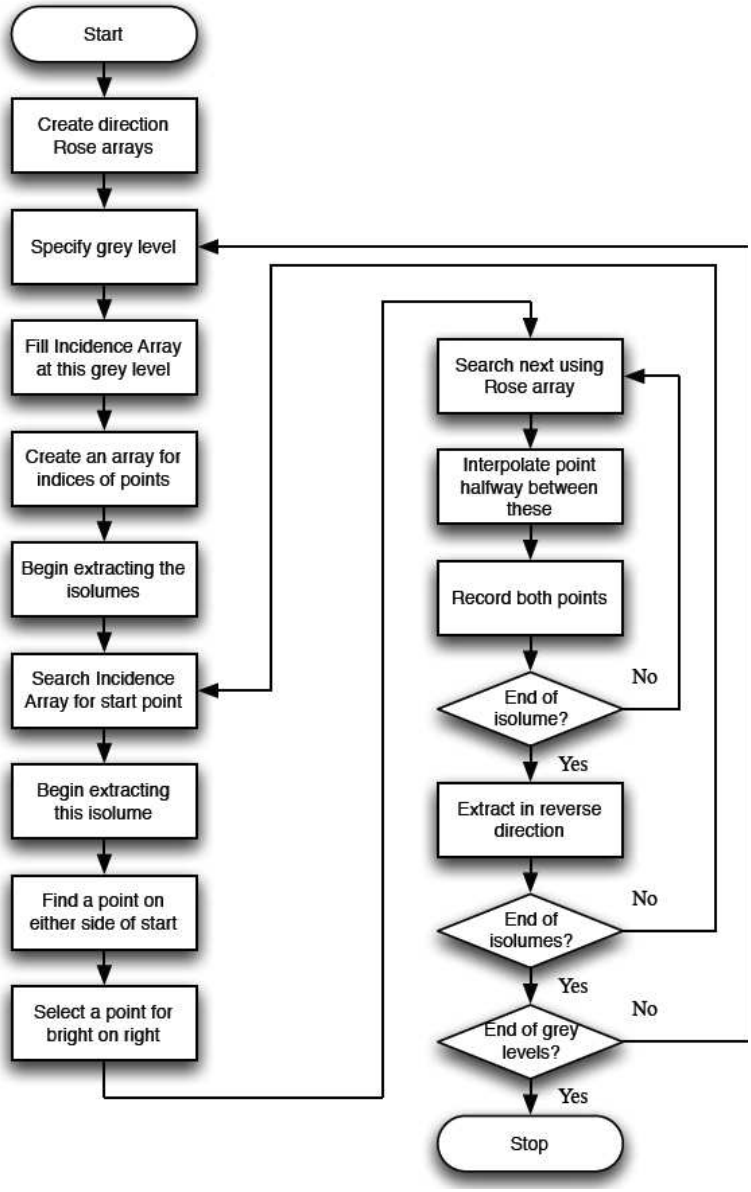


Fig. 6. Isolume Extraction Algorithm

But, before we can get to the elegant representation of the fingerprint (Fig. 7c), we have to manipulate the data quite carefully. A plot using the raw data is shown in Fig. 7b. If the data is simply averaged with five-point Gaussian smoothing, information is lost as everything is smoothed. We found that what was required was a filter that had a varying frequency response, depending on the signal. After considerable experimentation, we introduced two innovations. Firstly, in order to get the local curvature of the line, C, defined by:

$$C = d\psi/ds \tag{1}$$

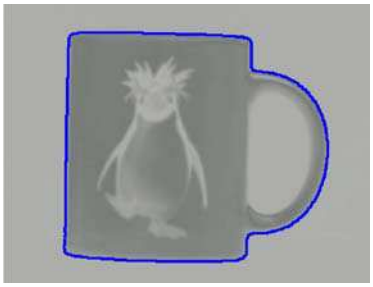


Fig. 7. (a) Isolume at outer edge of mug, shown in blue, starting at lower left and proceeding clockwise

Fig. 7. (b) Unsmoothed fingerprint of the isolume in (a)

Key: Lobe ☆ Line — Arc ⌒

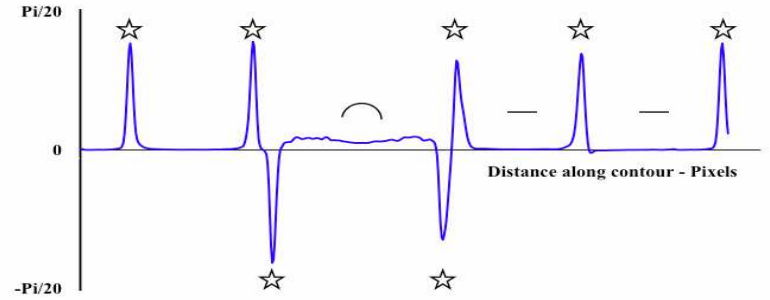


Fig. 7. (c) Dynamically smoothed fingerprint

where ψ is the angle of the local tangent to the curve and s is a measure of distance along the curve, we recognise that curvature is defined as the reciprocal of the radius of curvature. This naturally implies that it is the reciprocal of the radius of an arc which locally approximates the curve. It is then a very rapid calculation to fit a circle to three points on the data, spanning the central point at which the curvature is sought.

The second innovation was to vary the symmetrical distance between the central point and the two on either side from which the circle was computed. This distance has to be short where the isolume is turning rapidly such as the portion where it turns the corner at the top left hand corner of the mug in Fig. 7a, i.e. it is necessary to fit a small circle to capture the sharp rotation. However, as the isolume goes up the vertical side of the mug, it needs to be very large otherwise any imperfection in the isolume (arising from noise) gives rise to spurious values for the curvature. In this section of the fingerprint, it is clear that there should be no deviation from the straight line, i.e. we want a very large radius. We therefore wrote a routine to specify this distance between the centre point and the two outliers on the circle. We fitted a best-fit line to seven consecutive points, centred on the point in question. We then moved away from the central point and found empirically the distance, Span, for which the best fit line diverged from the isolume by an amount equal to (Span/10 + 0.5) pixels. It must be recognised that the coordinates defining each point of the isolume are precise to a fraction of a pixel but, as small differences are sought to determine curvature, this leads to large errors. Fitting a circle over appropriate distances ameliorates the difficulty. However, it should be recognised that the fingerprint Fig. 7c is not a perfectly accurate representation of the curvature at each point; it has lost something in the smoothing. But, since it is consistent and its distortion is not excessive, it is still useable. Once curvature of all points on the isolume has been determined, the data are all manipulated to give a set of points at one-pixel intervals with curvature, position coordinates and a precise distance from the beginning of the isolume associated with each point.

2.4 Extraction of features

The three types of features (lobes, lines and arcs) in the isolume fingerprint have certain characteristics. The extraction of each feature and a description of the characteristics are discussed below.

2.4.1 Extraction of lobes

The extraction of lobes follows the flowchart shown in Fig. 8. The smoothed fingerprint of the isolume is scanned to find groups of points with large curvature, C, and the local maximum curvature, C_{max}, representing the apex of the lobe. The area, A, under the lobe corresponds to the angle through which the isolume contour has turned as it passes along the lobe. Referring to the isolume around the outside of the mug in Fig. 7a, the first lobe corresponds to the turn of the contour at the top left edge through 90 degrees (1.57 radians) and this is the area under the first lobe in Fig. 7c. The second and third lobes would have similar areas, although the third lobe would be negative (an anticlockwise rotation through 90 degrees).

In addition to the area, we have defined, for lobes, two further characteristics, namely skewness (S) and kurtosis (K). Skewness is a dimensionless measure of the symmetry of the lobe about its centre and quantifies the extent to which the lobe is skewed to the right or left.

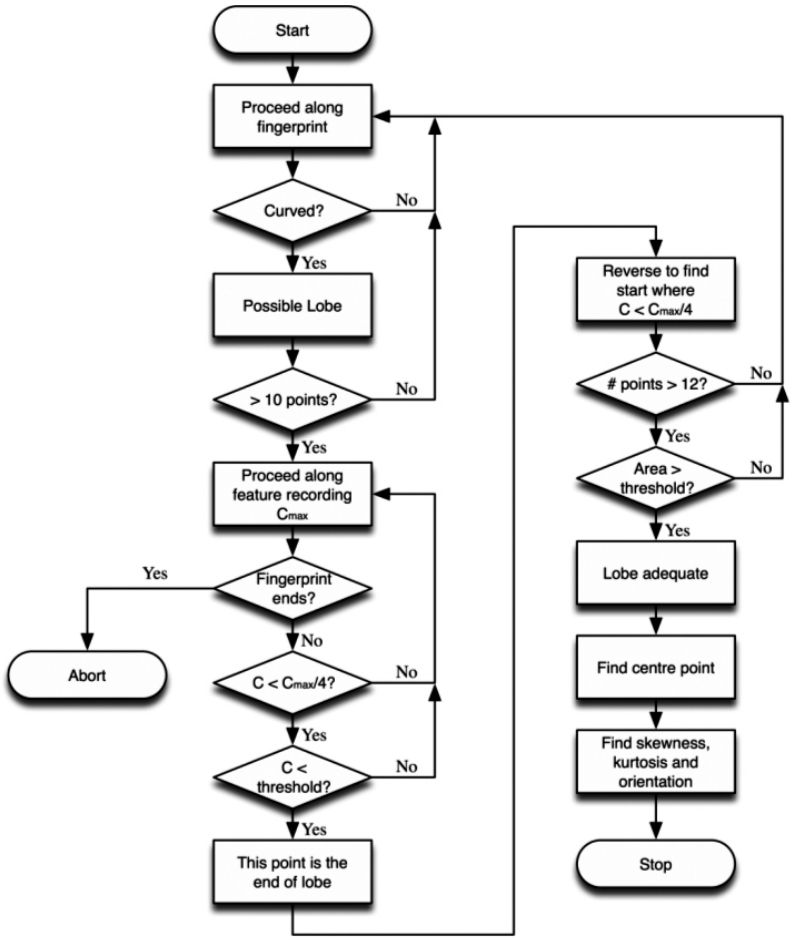


Fig. 8. Flowchart showing extraction of lobes and determination of lobe characteristics

It is defined as:

$$S = (C_G - M)/L \tag{2}$$

Where C_G is the centre of gravity of the lobe (in pixels), M is the mean of the greatest and least pixel values of the lobe, and L is the length of the lobe in pixels. For a symmetric lobe, $C_G = M$, so its skewness value is zero.

Kurtosis measures the extent to which the lobe is sharply peaked or flattened.

Let the lobe have curvature, C_i at the i^{th} pixel and let C_{max} be the maximum curvature of the lobe. Then the normalized curvature at the i^{th} point, c_i is:

$$c_i = C_i/C_{max} \tag{3}$$

The lobe starts at i_{start} and ends at i_{end}. Compute distance, W, such that W is the smaller of C_G-i_{start} and i_{end}–C_G. W is then the maximum interval of i, symmetrical about C_G, which is

contained within the pixel values of the lobe. Set n = 2W+1. Then compute a dimensionless second moment of area, M_1, about the centroid, summed over all i from C_G-W to C_G+W as:

$$M_1 = \{\Sigma c_i(C_G - i)^2\}/n^2 \quad (4)$$

Let c_{ave} be the average of c_i for the interval and compute M_2 over the same interval as M_1, where:

$$M_2 = \{\Sigma c_{ave}(C_G - i)^2\}/n^2 \quad (5)$$

Kurtosis, K, is defined as:

$$K= M_1/M_2 \quad (6)$$

Note that kurtosis is non-dimensional and that if c_i were constant over the interval (i.e. the fingerprint were linear as it followed a uniformly circular arc), then $c_i = c_{ave}$ and $M_1 = M_2$ and the lobe would have K = 1.
Fig. 9 shows two examples of lobes and their area, skewness and kurtosis values.

Fig. 9. Two lobes: Left lobe A = 0.82 radians, S = 0.05, K = 0.53
Right lobe A = 0.82 radians, S = -0.70, K = 0.32

2.4.2 Extraction of lines

Lines are represented in the fingerprint by portions of the fingerprint where divergences of curvature from zero are small. In real fingerprints, there will generally be some divergence and we need to decide what can be tolerated. Our strategy is to walk along the fingerprint until a suitable number of points is encountered that have very small curvatures, indicating that we are now on a line. Then we walk along the line and record a bad vote for every point whose absolute curvature is larger than some threshold. The votes would be incremented as bad points as they are sequentially encountered. However, as soon as an acceptable point (with curvature less than the threshold) is encountered, this sum of bad points is set to zero. When the bad point vote exceeds some threshold, the line is declared to be terminated and, provided that there are enough 'good' points (i.e. the line is an acceptable length), its centre point is recorded, its length is recorded and its orientation relative to the i-axis, in radians, is recorded.

2.4.3 Extraction of arcs

This is similar to the extraction of lines except that there are provisions ensuring that the values within the arc do not differ from each other by more than some normalised threshold.

2.4.4 Data management

Shrink-wrapped code was developed to examine an image, extract its data and to represent it as indexed isolumes containing sequentially indexed features, be they lobes, lines or arcs. (This code is accessible from the URL given in the appendix). By this artifice, the data of an image is compressed down to something of the order of 20 Kbytes. An alternative approach is to work solely from the features. Generally a particular feature is represented in several isolumes and it is of value to obtain a smaller list of super-features, each such super-feature being the average of its contributors – from several different isolumes. Each super-feature has data covering its exact position on the image and its properties, which would include its type (whether lobe, line or arc) and further appropriate specifications. A brass image (refer to section 1.4) might have 60 super-features, a gold image perhaps 45. The attributes derived for each feature are listed in Table 1.

Feature	Properties
Lobe	Area, Skewness , Kurtosis, Orientation, X_{centre}, Y_{centre}
Line	Orientation, X_{centre}, Y_{centre}
Arc	Radius, X_{centre}, Y_{centre}, $X_{arc\ centre}$, $Y_{arc\ centre}$

Table 1. Feature Properties

3. Object Recognition: the problem defined

The object recognition problem is now patent. We 'know' up to 15,000 objects. For each object, we have 20 views, each seen from one of the axes of an icosahedron. This gives up to 300,000 views (or gold images) as our knowledge base. As we take an arbitrary view (a brass image) of any particular object, we can at most be twenty one degrees away from one of our cardinal views and we expect to be able still to recognise any object, even with some small difference in orientation.

Our intent is to deploy AV to run, unsupervised in an autonomous robot. The robot will acquire information during the looking process that will isolate objects one from another and from the background. It will look at some portion of its world and perform stereopsis in order to get a measure of distance to points in its view. It assumes that objects are sui generic in distance. This is to say that the dimensions of the object are small compared with the distance to the object from the eye. This is generally so, with exceptions such as when objects are brought very close to the camera. The robot's artificial intelligence (AI) categorises areas in its view as to distance and when it finds that a set of points is of similar distance, distinct from the background, it assumes that they represent an object. It might also be guided by coherent movement of the points against the background as a result of the camera moving or the object moving. With this information, it isolates some portion of its field of view. This is now termed the 'brass' view, possibly containing an object. It then enquires whether any one of the 300,000 object views is present. These 300,000 views contain of the order of 100 isolumes each. If we condense multiply represented features down, we will have something like 100 super-features per view, i.e., 30,000,000 super-features. This is not intractably large but neither is it a simple problem because we do not have an ordering principle. Further, much of the data is non-digital in the sense that the area of a lobe is an imprecise number which may be slightly different every time we measure it in a different

image. Therefore the technique of hashing which is of such power in many database searches is not readily available to us.
We have tried three approaches to the problem.

4. First approach – Matching of features

Note that we are here working exclusively with super-features - those features which are multiply represented in different isolumes at a point. This approach requires an initial search through the 300,000 views of the gold image database to produce a list of views which have features in common with the brass view. The resultant list will be ordered in terms of feature commonality. It is hoped to reduce the candidates by a factor of 10,000 in a coarse separation. With this reduced list, we can then enquire sequentially, as to the probability that each view on the list is actually a match with the brass image.

4.1 Coarse search

Mining of data from the database relies upon choosing a suitable key or index that can be used to extract useful candidate records very quickly from tens of millions. Speed of retrieving records is as important as the requirement that not too many possible matches be missed. We search only for matches of lobes on the basis of Area, Skewness and Kurtosis (ASK). Lines and arcs are not considered in the first instance because they are not as well defined as lobes.
Searching a database for records that contain the correct values of these discriminators to within a suitable interval is inefficient because hashing and binary searches are not possible. A better approach would be to sort them into bins and search for a coding representing membership of a particular bin. A search through the database for candidate features now amounts to looking up an index for matching bin numbers. This can reduce the search time by orders of magnitude.
The choice of the width of the bins is important since bins that are too narrow will result in measurement errors placing discriminators in the wrong bin. Conversely, bins that are too wide will decrease discrimination. The optimum size appears to be the same as the expected maximum measurement error. Where the measurement error is a percentage of the measurement rather than a fixed number this will naturally lead to the bin sizes increasing with the size of the discriminator; the log of the bin width will be a constant. We refer to these as log bins.
With log bins there is an infinite number of bins between any given value and zero. One 'catch-all' bin can be included to span the range 0 to the smallest log bin with the rest of the bins spanning the range to the largest bin.
There is a further concern however. Regardless of the size of the bin, it is possible for the discriminator to be placed in the wrong bin because it is too near the bin's edge and there is variation in the measurement from one image to the next. This can be overcome by considering membership to include not only the given bin but also the measurement's nearest neighbour bin. This will suggest that, with the bin no narrower than the measurement error, a search of the database will turn up most relevant candidates.
Database search engines are capable of concatenating search indices, so that all three discriminators can be searched simultaneously. Unfortunately, including the nearest-neighbour bin requires eight separate searches be undertaken, one for each of the eight possible combinations of two possible values (one bin and its nearest neighbour) of the three discriminators. Because the search is definite, it is fast, notwithstanding the extra bins.

Generally an object seen in a brass image will be rotated, in the plane of the image, relative to the gold image. Therefore two other relationships can be used. These involve the individual angles and distances between features in a group. (Recall that each lobe has an orientation angle relative to the image.)
If we plot all the lobes of our database in ASK space, we might expect them to cluster, with the population density declining with distance from the centroid of the cluster. We can determine the variance of this distance over a large population and by using a cutoff, in our initial search, we concentrate on lobes which are far from the centroid, i.e. abnormal; they will offer greater discrimination.
With these ideas in mind, we applied the following process:

- Find candidate views
- Discriminate geometrically
- Perform a more careful discrimination on the small number of remaining candidate views by:
  - clustering
  - iterative correspondence

4.2 Find candidate views

We find a list of candidate gold views by searching for the features of the brass image in the gold image library. We use only lobes and choose only those brass lobes which are abnormal (as defined above). This involves searching a database of up to thirty million gold features for matches to 25-75 brass lobes. We use only lobes because they have more discriminators than lines or arcs.
The resulting list of candidate views is ranked according to the number of lobes which match between gold and brass. The gold view which has the greatest number of matching brass lobes is at the top. Now we have to ask whether this best view actually matches the brass view. If not, we will consider the second best gold view and so on.
The first step is to produce a matched list of gold and brass lobes. Conceive of a list of gold lobes on the left, matched with brass lobes on the right. This is made up by taking the first gold lobe and then scanning the brass lobes to find the best match. A best brass match will have discriminators that best match the gold lobe. If they don't match well enough, then we will discard this gold lobe. Once we have the pared down list, we need to find out how many of these matches are in fact 'true.' Does the gold lobe really correspond to the brass lobe as we look at the images? Of course the comparison algorithms can't 'look' at the picture to see matches but at the end of the procedure, we can determine how well it functions by examining the images ourselves.

4.3 Geometric discrimination

The objects in the two images will generally be rotated with respect to each other by an angle, θ, and have a dilation factor, δ, i.e. the gold object will be bigger or smaller than the brass object. For those gold features which are in the brass view, they should all be on the same object and therefore they should all be turned through θ. In fact, the difference in orientation between the brass and gold feature should be equal to the rotation angle, within our measurement uncertainty limits. This can be seen in Fig. 10 where θ is the difference between the orientation of feature, f1, in the gold image and f1' in the brass image and also between f2 and f2'. The orientation of the features is shown by the arrows.

The angle between features can then be tested, i.e. the angle of the line joining a given pair of features in the image. For each connecting line the difference in angles should be θ, the rotation angle.

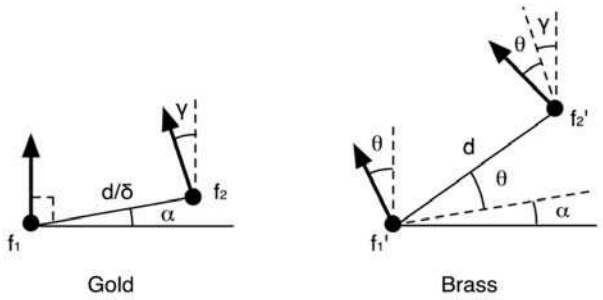


Fig. 10. Relationship between features in gold and brass images.

In a similar fashion we can calculate the ratio of distances between any given two features in the brass image and the distance between corresponding features in the gold image. The ratio should be the dilation factor.

Lobes have an orientation and a position in the image. Lines and arcs, even though not as well defined, can still be used because we can get the shortest distance between a line and a lobe or arc and we can use the line orientation. Arcs do not have an orientation but the radius can be used as an initial filter and the angle and distance to other features can be used.

Having calculated our measure of geometrical similarity for a given gold/brass pair of features, the task is now to eliminate the mismatched feature pairs. Recall our two lists. On the left is a list of gold features; on the right a list of brass features. We assume that the first gold feature corresponds to the first brass feature etc. This is on the basis of discriminator matching. But some of them will not correspond; the feature referred to in the left list will not be the same part of the object as that in the right list. We have considered two different methods to deal with this; clustering and iterative correspondence.

4.3.1 Clustering

Clustering accepts all the table pairings and then discards those pairs which do not correspond. Consider the rotation angle, θ, as the difference in orientations of brass and gold feature of each pair. If the gold view matches the brass view, and the pairs we have chosen are mostly correct, we should find that many of them have a constant difference in orientation equal to θ. We can find this angle and eliminate bad pairing by iteratively removing outliers from the cluster. The average of all the differences in orientation is found and assumed to be θ. The deviations in differences are taken from this and the feature pair with the largest deviation is struck out. The process is then repeated until the largest deviation falls within acceptable limits or we run out of feature pairs.

We can then look at the line features in both views, although lines were not included in our matched lists. By their nature, the orientation of lines is far more precise than that of other features. All possible line/line matches are examined and those that are sufficiently close to θ are used. These are then clustered in the same way to achieve a better estimate of the rotation angle, θ.

Now we can perform the same type of clustering operation on the dilation factor by considering the ratio of inter-feature distances between the gold image and the brass image.

This method is quick but suffers from the fact that we cannot improve on our initial matching of the two lists. We can progressively discard bad matches and we will generally fail when we don't have enough matches left. Alternatively, if the gold and brass views agree, we will see it because we end up with a reasonable number of matches.

4.4 Iterative correspondence

To overcome the deficit of poor initial matching, we could consider all possible matchings. For 75 features in each image (i.e. the brass image and the gold image) there are of the order of 75^{75} possible combinations, a number not to be seriously considered. (Note that $75^{75} = 10^{140.}$ This is huge compared with the 10^{79} electrons in the universe.)

There is a more efficient process. Choose the first gold feature as the pole feature and then sequentially attempt to match each brass feature with it. If the discriminators match to acceptable accuracy, consider the next available gold feature as the second feature. Run through the remaining brass features until a match is found in terms of discriminators. Now test geometrically to see whether the rotation of the second pair matches the rotation of the first pair. If it does, seek a third pair and apply tests for rotation plus dilation. Proceed in this way until a match cannot be found for some n^{th} feature of the gold list. In that case, revise the $(n-1)^{th}$ feature and proceed. In principle, this would entail the same number of possible combinations but in practice, when a certain initial match is discarded, this discards all the possible consequences and the method becomes quite quick.

Furthermore, since the geometry tests are all symmetric we need only test half of these possibilities. And, finally, the order in which the pairs are tested is not important. If a set of feature pairs, abc, is a successful combination, then so will acb or cba. This reduces enormously the potential combinations to be searched.

There is one exception to this, which is the choice of the first pair-pair combination. This is used to determine the initial estimates of rotation angle and the scale factor. Since the geometry tests are passed or failed on a tolerance figure, we choose to assume (with some claim to validity) that any truly matching features will adequately represent scale and rotation.

As the number of successfully-tested feature pairs increases, so do the number of combinations to be tested at each next step but, beyond a certain level, the possibility of the brass and gold images not matching becomes insignificant. Early testing has suggested that limiting the number of tests to 3 times the number of feature pairs squared is quite sufficient ($3 \times 75^2 = 16{,}875$).

All the feature pairs that remain have passed scrutiny. Their number will be the most robust indicator of whether the objects in the two images are the same.

4.5 Implementation

The coarse search of section 4.2 was implemented in MySQL, interrogated from C#.

A speed test was conducted by filling the gold database with three million features and performing searches. The characteristics of the features were randomly assigned. A brass image with 81 features was used. These 81 features were then searched for in the RAM-resident gold database of three million features. The ASK search key was employed, using eight searches to handle the binning problem of ASK. In a mid-line 2007 desktop computer the search took about 100ms. This implies something like one second to search the known universe of thirty million features, which is quite satisfactory because this time will halve every two years as computers improve.

4.6 Results: Object recognition based on matching of features

It was found that iterative correspondence was better than clustering. The results for object recognition based on matching of features with Iterative Correspondence are shown in Table 2a for the mug and Table 2b for the measuring tape.

Table 2a shows that the mug was recognized in all 14 of the brass images where it was unoccluded but that recognition success dropped rapidly the more the mug was occluded by other objects. 'False negatives' are the number of brass images which contained the mug but which the technique failed to recognize as the mug. 'False positives' refer to those brass images which did not contain the mug although the technique erroneously found a mug. In fact, there were no false positives. 100% occlusion means that the object is not present in the brass image.

Table 2b shows the results for the measuring tape.

Occlusion (%)	0	<25	<50	>50	100	Total
Images	14	13	5	2	66	**100**
Successful	14	8	1	1	66	**90**
False negative	0	5	4	1	0	**10**
False positive	0					

Table 2a. Results – object recognition based on matching of features for the mug by Iterative Correspondence

Occlusion (%)	0	<25	<50	>50	100	Total
Images	28	4	7	0	61	**100**
Successful	21	3	2	0	61	**87**
False negative	7	1	5	0	0	**13**
False positive	0					

Table 2b. Results – object recognition based on matching of features for the measuring tape by Iterative Correspondence

5. Second approach – Triples

5.1 Introduction

As one searches the database for features only, much of the geometric information inherent in an image is not considered since the features do not contain any information about their relationship to other features. This is considered to be a flaw in the previous approach where the geometric information has to be considered afterward. To rectify this, the concept of *triples* is introduced. As suggested by the name, these are triplets of features. The triples are created as every possible combination of three features.

The total number of triples created with n features is given by n(n-1)(n-2) /6. If n is 20 this results in 1,140 triples but, since this increases with order 3 as n increases, a practical limit is reached fairly quickly. At 40 features we have 9,880 triples, which is approaching the reasonable limit for an image. While this is a large number it should be remembered that gold images do not need to have as many features/triples—since the object will be in relative isolation—and will, therefore, pose a smaller burden on the database.
Each triple can now be considered as a triangle on the image with a feature at each vertex. This leads to a number of intrinsic geometric properties that are scale and orientation independent. This is delightful since it allows us to analyse any image for similar triples without regard to the size or orientation. We can extract several measures from each feature.

5.2 Triple orientation

The orientation of a triple can be uniquely defined in many ways. We choose to define the axis of the triple as that directed line that bisects the shortest side of the triangle formed from the three features and passes through the opposite vertex (Fig. 11).
This provides the most accurate measure of orientation. The vertex thus bisected is called the *top* of the triangle and is considered to be the 1st feature in the triple. The 2nd and 3rd features have the median and largest sides opposite them respectively (Fig. 11).
While the triple orientation is not a rotation-independent parameter it is very useful to us in deriving the following parameters that are rotation-independent, as well as providing a useful parameter to use later in discovering the rotation angle between the images.

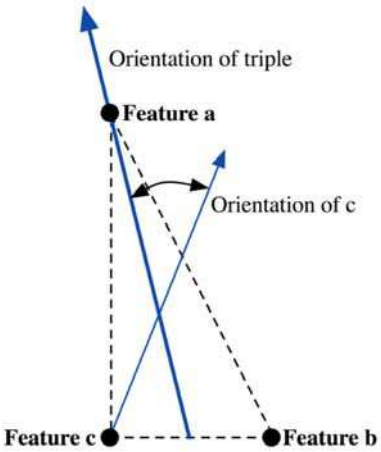


Fig. 11. Triangle formed from three features in an image. The triple orientation is given by the line bisecting the shortest side and passing through the opposite vertex. Feature orientations are given relative to this line.

5.3 Maximum and minimum distances

If one takes the longest and shortest sides of the triangle and divides them by the average length of the three sides one will have two numbers that are scale independent and which code for the shape of the triangle.

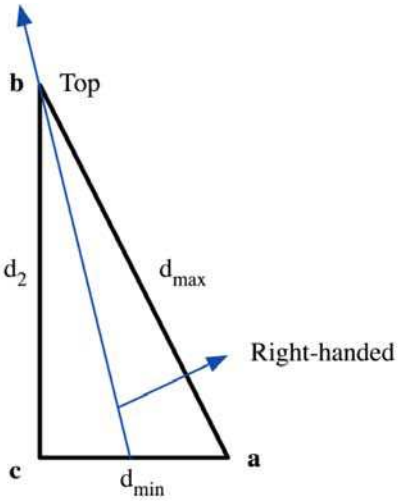


Fig. 12. Triple showing the definition of the top vertex and handedness.

5.4 Maximum and minimum angles

Since the internal angles of a triangle will sum to 180° we can describe the shape of the triangle with only two angles. These are chosen as the largest and smallest angles. They have the important property that, like the shape of the triangle, they are scale- and rotation-independent.

These two angles have an advantage over using the distances in that, for a flat triangle, the shape is highly sensitive to the distances but not to the angles.

5.5 Triple handedness

The parameters of the triple already mentioned will constrain it completely, apart from its chirality; it will appear identical to its mirror image. To break this symmetry we can define a *handedness* for the triple as the side of the orientation vector on which the longest side lies when the orientation vector points directly upwards. If the longest side is then on the right, it is a right-handed triple (see Fig. 12).

5.6 Maximum and minimum relative orientations

The relative orientations of the three features at the vertices of the triangle provide orientation- and scale-independent parameters. In this work the relative orientations are defined with respect to the orientation of the triple, i.e. the relative orientations are the clockwise angle between the triple's orientation and the feature's orientation.

Note that all three are independent of the shape of the triple's triangle.

5.7 Disambiguation

Since triples are to be matched with regard to their shape, it is important that there should be no possibility of ambiguity. For instance, were two lengths of a triple similar to each other, it is possible that in some images, one would be measured as the longer and in other images the reverse might occur. Accordingly, only those triples are accepted for comparison

where there is a difference of at least 10% in the three lengths. Although this discards some triples, those remaining are always unambiguous.

5.8 Comparison strategies

We assume that we have done a coarse match, as described in section 4.2 and have winnowed our 300,000 views down to a small number, ordered in terms of probability. Our task is once again to hold up a gold view against a brass view and to determine whether they are the same. There may be about 10,000 triples in the gold view and perhaps 20,000 in the brass view. Depending on the makeup of the triple, i.e. lobe-lobe-lobe or lobe-line-arc etc., the triple will have up to 16 discriminators associated with it. To compare them sequentially and for each discriminator implies up to 3,200,000,000 tests. Many of these are comparisons for matching within a threshold rather than simple checks for equality. Consequently, on the face of it, this method will be computationally very intense.

On further consideration, we see that, if we order the tests in order of discrimination, we will quickly cut down possibilities. Experimentally we found that, if the test for equality of largest angle between triples was run first, this cut out 19 out of 20 contenders and as we followed this by other stern discriminators, the total fell very rapidly. Secondly we note that such an overall comparison can readily be done with parallel processing. Since this technology is being progressively deployed, the problem will become steadily more tractable. Currently, it is completely feasible to deploy the problem on an NVIDIA processor with 256 parallel cores and it is unlikely that this will be the high water mark for hardware. Thirdly, it is possible to sort the triples on the basis of a discriminator and to perform a binary search to the upper and lower limits of acceptance of this discriminator and then do a detailed comparison for those triples within these limits. In this way, 20,000 tests can be reduced to about 100. So, although this method is computationally intensive, this need not be its death knell.

As a first step, we can produce a list of triples which satisfy the sixteen discriminators for lobe–lobe–lobe triples (and the somewhat smaller number for other triples). But these matches are not necessarily true so we need to introduce further geometrical information to eliminate bad matches. For this the angle of rotation between the images and the dilation are needed. We used the method of clustering described above in 4.3.1. At the end of this procedure we are left with only those triples that have been matched with regard to all the discriminators and have the same orientation and size relative to the gold image.

5.9 Global application

We applied this technique to all the triples in the gold image and ran them against all the triples in the brass image, without regard to the time taken for the comparison. Our results seemed to indicate that performance was dominated by threshold settings in the comparison of discriminators; there were just too many possible solutions and inevitably each gold triple would match too many spurious brass triples. Avenues for advance were still open in that tests for orientation and dilation could narrow down the huge list of possible matches. But we elected to abandon this approach in favour of a more selective criterion.

5.10 Matching of isolumes using triples

By the nature of the process by which they are generated, isolumes provide strong organisation of all the features in the image; those features which appear on an isolume have an enduring relationship with each other. If, therefore we only compare all the triples on a gold isolume against all the triples on each successive brass isolume, we can expect to reduce

the number of possible combinations. We would expect the ratio of matches for matching isolumes to be very much larger than for unmatching isolumes. In this enterprise, we are not using aggregated super-isolumes but merely individual isolumes.
First, we perform the coarse search of 4.2. This results in a short list of gold views which might match our brass view. We then hold up each gold view against the brass view to see if it matches. In this process, we run the triples of the first isolume against the triples of each brass isolume and so on through the list of brass isolumes. Then we do this for the second gold isolume etc.

5.11 Results

The results for object recognition based on feature triples (discussed in section 5.10.) are shown in Table 3(a) for the mug and Table 3(b) for the measuring tape.

Occlusion (%)	0	<25	<50	>50	100	Total
Images	14	13	5	2	66	**100**
Successful	14	10	2	1	63	**90**
False negative	0	3	3	1	0	**7**
False positive	3					**3**

Table 3(a). Results for object recognition based on feature triples for the mug.

Occlusion (%)	0	<25	<50	>50	100	Total
Images	28	4	7	0	61	**100**
Successful	28	2	7	0	60	**97**
False negative	0	2	0	0	0	**2**
False positive	1					**1**

Table 3(b). Results for object recognition based on feature triples for the measuring tape.

5.12 Discussion

It can be seen from a comparison of the totals in Tables 2 and 3 that the introduction of triples has significantly increased the ability of the system to identify the gold objects, at a cost in false positives. In particular, the recognition of the unoccluded tape has been raised to 100% (compared with 75% for object recognition based on feature matching) as well as raising the recognition of the occluded tape from 45% (5 out of 11 images) to 82% (9 out of 11 images).
On this basis, and remembering that these results are for a fairly small library, this method has shown the potential to be used for generalised object recognition.

6. Third approach – Isolume matching

6.1 Introduction

Analysis of an image produces a number of isolumes and a set of features threaded on the isolumes. The first approach, enumeration, viewed the body of data as being the features, each with attendant descriptors, but essentially unrelated to each other until a match had been suggested. At this point it was feasible to introduce geometrical relationships between features in order to confirm the match. The second approach, triples, introduced arbitrary matchings of three features and embodied the geometrical relationships between them. This approach produced a very large number of triples which had to be compared. The new scheme, Isolume Matching, seeks to reduce the scale of the search by using the intrinsic ordering of the features by the isolume. Their order on the isolume is fundamental information and can be used advantageously.

With this in mind, we looked at the efficacy of matching contours on the basis of the order and properties of the features. It is immediately apparent that the possible scope of this work is large because we might aim at producing a robust search that would not stumble if a feature were missing either in the gold or the brass isolume. It would also tolerate a spurious feature in either isolume. But, at the outset, we consider a simple search that demands only that two isolumes be deemed to be matched when the order of the features is identical and corresponding properties match to within some threshold. Again, we are certain that later workers will massively refine our crude first efforts – provided that we show them to have merit.

We are operating, after the coarse search of section 4.2, on a candidate gold view and the question is whether this matches the current brass view. Typically the gold view will have up to 100 isolumes, each with ten or twenty features. The brass image will probably have slightly more isolumes, say 150. It is necessary to compare each gold isolume with each brass isolume – 15,000 comparisons. We do not demand that the isolumes should match over their entire lengths. It is probable that a match over quite a small number of features will be significant. Each feature must match a number of attributes that will depend on the feature. Lobes must match four attributes, including type.

Given the level of precision of these attributes we estimate that we can distinguish an unknown lobe against a known lobe to a discrimination of one part in 480. This opinion derives from discrimination to a factor of 3, based on feature type (lobe, line, arc), a discrimination of 10 based on area, 4 based on skewness and 4 based on kurtosis. These crude factors derive from our perception of the accuracy of the measurements. Lobes generally outnumber other features in most images by about 2:1. Lines provide a discrimination of one in three – solely on type. Arcs provide the same discrimination. We estimate that when we obtain five consecutive matching features, this generally provides a possible discrimination of the order of one part in a billion (say, three lobes and two non-lobes, i.e. 480x480x480x3x3). Of course this takes no account of the distribution of properties and, in practice our discrimination will be much less efficacious. But, once we have the crude match, it is likely to be true and, it is profitable to go to a more careful match where we can look into the geometrical relations between the five features. This latter, time-intensive comparison will only occur for very well-screened candidates.

6.2 Structure of the search

The logical structure of the search is simplified by introducing the concept of a cardinal feature. This has a different value for gold and brass. Call them $G_{Cardinal}$ and $B_{Cardinal}$. Then

write a function called Compare(, , ,), which compares two features, one gold and one brass. The argument list of the function includes the values of the cardinal numbers and also includes D_{Gold} and D_{Brass}. These latter numbers indicate by how much we want to increment the cardinal value. For instance, an argument list of Compare($G_{Cardinal}$=1,$B_{Cardinal}$=5 ,D_{Gold} = 1,D_{Brass} = -1) would imply that on the isolumes in question, we were comparing feature number $G_{Cardinal}$ + D_{Gold} = 2 on the gold isolume with feature number $B_{Cardinal}$ + D_{Brass} = 4 on the brass isolume. This shorthand makes it very easy to keep track of five sequential comparisons in a nested IF structure. Clearly, as we lay feature number $G_{Cardinal}$ opposite feature number $B_{Cardinal}$, we can then sequentially compare these and the next four features by setting D_{Gold} and D_{Brass} from 0 through 4. Such a nested structure is simple to program. It is also relatively easy to check for the case where the order is reversed, in which case the values D_{Brass} would have opposite sign. As we become more sophisticated, this structure will lend itself to considering occluded and spurious features in either the gold or brass isolume. For the moment, we elect to use a five-feature check. The optimal value for this number must be determined by experiment in the fullness of time.

For the search, we set up two loops, running each gold isolume against each brass isolume. This is shown in Fig. 13.

Within each such confrontation between isolumes, we run sequentially through all the features of the gold isolume, setting each feature as $G_{Cardinal}$. For each of these features, run through all the brass isolume features setting each as $B_{Cardinal}$ and then running forward for up to five features. The test will almost always fail after one or two features and we can increment $B_{Cardinal}$ until we reach the end of the isolume. If it fails after one forward test, it attempts to match them in reverse order. If we get five matches, we do a 'Compare-in-Detail' test.

6.3 Compare-in-Detail

Assume that we have five sequentially matching features. We can then do further tests;

- Find the distances from the first non-line feature to each following non-line feature. Sum them and divide each of the above distances by this sum. The resulting numbers will be Scale- and rotation-invariant. Compare them severally with their brass equivalents. Then find that non-line feature that is farthest from the first non-line feature determined above. From this remote feature, find a set of distances to each non-line feature and normalise them using the above sum. Compare the equivalent values for brass and gold. This has the effect of taking a cross-bearing and demanding that all non-line features are in the same geometrical relationship to each other for gold and brass.
- For any arcs, normalise the radius with respect to this sum and compare between gold and brass.
- For lines, determine the angle of rotation between successive lines among the line features and demand that these angles be the same for gold and brass. This test will generally discriminate against mirror images.

After the initial match of five features and following it by the above protocol, the level of specificity is very precise and we can declare that the two portions of isolume do indeed match.

6.4 Evaluation of the method

We ran gold views of a mug and a tape against the hundred images of our database. For a particular gold view matched against a brass view, we ran perhaps 100 gold isolumes against

```
Hits = 0
For Gtrace = 1 To # gold isolumes
For gCardinal = 1 To # lobes this isolume
   For Btrace = 1 To # brass isolumes
      For bCardinal = 1 To #lobes this isolume
         If Compare(gCardinal, bCardinal, 0, 0) Then
            If Compare(gCardinal, bCardinal, 1, 1) Then
               If Compare(gCardinal, bCardinal, 2, 2) Then
                  If Compare(gCardinal, bCardinal, 3, 3) Then
                     If Compare(gCardinal, bCardinal, 4, 4) Then
                        If CompareInDetailForward(gCardinal, bCardinal) Then
                           Hits = Hits + 1
                           Goto BailOut
                        End If
                     End If
                  End If
               End If
            Else
               If Compare(gCardinal, bCardinal, 1, -1) Then
                  If Compare(gCardinal, bCardinal, 2, -2) Then
                     If Compare(gCardinal, bCardinal, 3, -3) Then
                        If Compare(gCardinal, bCardinal, 4, -4) Then
                           If CompareInDetailBack(gCardinal, bCardinal) Then
                              Hits = Hits + 1
                              Goto BailOut
                           End If
                        End If
                     End If
                  End If
               End If
            End If
         End If
      Next bCardinal
   Next Btrace
Next gCardinal
BailOut:
Next Gtrace
```

Fig. 13. Algorithm for Comparing Isolumes

perhaps 150 brass isolumes. If we found a match between five sequential features in the two isolumes we were comparing, we then did a 'Compare-in-Detail' test and perhaps declared that the isolumes matched. We then moved on to the next gold isolume. This has the effect that, when we recognise an object, the search is quicker than when we do not. At the end of the matching process, we are left with a fraction of all the gold isolumes which had counterparts in the brass image. Note that we only demand a match over five features and, when we have this, we look no further. Even without optimizing the code, the comparison of a gold view against a brass view took under a second. It seems that on a 2010 mid-range desktop computer, we can compare a gold with a brass image in under 50 milliseconds.

It might be that the crude fraction determined above could be improved by expressing it as the number of five-feature sequences matched between the two images as a fraction of the number of five-feature sequences available in the gold image. We used only the first, crude, comparison strategy because we found, experimentally that it gave us a sharp discrimination. Since we are only looking at a small part of the isolume we could expect that the lower, rectangular portion of the mug would look like the lower, rectangular portion of a tape or, indeed, like the lower, rectangular portion of any rectangular object. Thus, when we ask, is the tape a mug, we will get a non-zero result because it is in fact a bit like a mug. However, if we ask, is an actual mug a mug, we will get a much more vehement result because there will be so many more isolumes that will match. The results are presented in Table 4.

Occlusion (%)	0	<25	<50	>50	100	Total
Images	14	13	5	2	66	**100**
Successful	14	6	2	1	66	**89**
False negative	0	7	3	1	0	**11**
False positive	0					0

Table 4(a). Results for object recognition based on Isolume Matching for the mug.

Occlusion (%)	0	<25	<50	>50	100	Total
Images	28	3	7	0	62	**100**
Successful	25	3	2	0	62	**92**
False negative	3	0	5	0	0	**8**
False positive	0					0

Table 4(b). Results for object recognition based on Isolume Matching for the measuring tape.

7. Evaluation of the three methods

Examination of Tables 2, 3 and 4 allows a crude comparison of the three methods. All of the methods provide clear and unambiguous recognition of unoccluded objects, with very few false positives. In fact, on the basis of the tables, there is little to choose between them. Their performance for occluded objects is also similar.

In terms of computational burden, the third method has a clear advantage. It also has the advantage of being conceptually more akin to the human recognition process. Finally, it seems to have more room for improvement than the others. We have deployed a very crude application and found very good results. Clearly, as we extend the application, as we have already done for the other two methods, we can expect a performance improvement. The isolume matching method also lends itself to very elegant general searches of the whole

database. We will not discuss this here but we can envisage very much more precise and quicker general searches than the coarse feature search discussed in section 4.2.
The tables of results were produced in order to provide a sense of the performance of the three methods for comparison. The tables make the statement that a particular object was or was not recognised in an image. But the biological experience does not deal in such certainties. All perceptions should be considered as probabilities. In fact, it is logically impossible in the biosphere to ascribe certainty to any event or condition because of the solipsistic argument. "But", you argue, "I am as sure as I need to be that this computer screen is on my desk." That is true, but if you were permitted just one glance, lasting a fraction of a second (this corresponds to the conditions which led to the above tables, where we are comparing one glance with a reality defined by the gold view), into a strange office, you could not make that assertion; it is only after you had verified the assumption two or three times that you could make the statement, and believe it, whether it were true or not. This is the human condition. And it must be the condition of a robot operating in the same circumstances.
As we set about producing vision for our embodied intelligence, we would be wise to structure our determination of reality in a similar way. We therefore need, not a decision as to whether the object is present in the picture, but a probability. This probability can only be determined by extensive experience of the method, predicting and comparing with reality. Consider that we examined a brass image and found 50% of the isolumes of a gold image to be present. In another brass image we might find 75% of them to be present. On this basis, we would certainly not be able to assign probabilities to the two findings. We would need to operate in the world and find the actual probabilities of independently-verified existence and then form a non-linear calculus in order to relate proportion of isolumes recognised with probability of existence. Even then, some objects might be more definitely recognised than others so that this functionality would have to be dependent on the object. Fortunately, we do not need to explore this concept at this stage.
As we consider using our method, the unavoidable problem occurs that certain objects are intrinsically similar and, as we deal with the variation within a universal classification, we can expect positive responses from many similar objects. It is our observation that we can be guided by the question we have asked the database, in the following sense. Not unlike the human perceptual system, we will operate essentially on the basis of perceptual hypotheses. Thus, guided by our coarse search, we always ask, "Is this gold view present in this brass image?" And we will get an answer couched in the form of the proportion of isolumes of the exemplar which we have recognised. But a lower value might be the result either of partial occlusion or else divergence in appearance between the object in the brass image and our exemplar. This uncertainty might be resolved by a further exploration. Imagine that there is a mug but no tape in the brass image. When we ask if a mug is present we get a matched isolume proportion of ξ. When we ask whether a tape is present, we will generally get a much lower but non-zero value for ξ. This is because both mug and tape have a rectangular lower section. Clearly, when we have explored all the probable options of what objects might be present, we will be either secure in our uncertainty or else have a clear judgement as to which possible object has an overwhelming ξ.

8. The Problem of Universals

8.1 The metaphysical problem stated

Plato (about 350 BC) was concerned with the theory of forms and presented his allegory of the cave where denizens could see only the shadows of objects. He argued that we see only

imperfect forms of the ideal unchanging forms; these alone are true knowledge. Thus the cup which we see, in all its varieties, is a corrupt exemplar of a perfect cup. We can see how Christian doctrine was not averse to this view. Diogenes of Sinope offered the refreshing observation that "I've seen Plato's cups and table, but not his cupness and tableness" (Hicks, 1925). Philosophers over the ages have weighed into this delightful debate which is not susceptible to proof but can be thoroughly discussed.

8.2 Universals in the world of AV

We argued above that there are some 15,000 nouns which might be known to educated persons. One of these is 'cup' but clearly there is a huge variation of objects – from tea cup to D-cup, all of which fall under the universal of cup. How shall we deal with this problem which, on the face of it, is very difficult? We hesitate to appear to be wiser than Plato, but can nevertheless offer a simple solution;

If we advance the proposition that all exemplars of a universal can be transformed into each other by simple physical distortion, our problem becomes really quite easy. Consider the tea cup which we can deform quite easily by barreling the sides and reducing the base, into a bra cup. If we couldn't do this, the English language, in all its whimsy, would not have called both of them 'cup'. This is a fairly profound philosophical statement that the ideal of cupness resides in our perceptions and we are prepared to use it on those objects which satisfy our criteria for 'cupness'. The fact that the D-cup does not have a handle and yet retains 'cupness' implies that the presence of a handle is not important. We think this puts us on the side of Plato, rather than Diogenes.

8.3 Application of universals to our method

Consider the universal 'car'. As we view cars from a distance and are not concerned, for instance, with their differing hood ornaments, we see a considerable similarity among them; enough for us to ascribe a universal name to them. This universal is not ascribed on the basis of function but of shape. We can see that, by a fairly clear and simple process of distortion, we can morph an SUV (sport utility vehicle) into a sedan into a sports car.

As we consider one of our twenty cardinal views of a car, we immediately perceive that there is a need to standardise these views so that all views from the top of the car are the same, whatever car it is. We can readily agree on the three Cartesian axes within which we would embed all cars, probably choosing the road surface as one of them. In fact, we seem predisposed mentally to assign cardinal axes to objects which have symmetries. There may even be a deep-seated inclination to view their structure in terms of quadripedal symmetry. We see the clear advantages, to our AV calculus, of viewing objects in this way. Let us say that we have the same view of two different cars. We assume that, based on our agreed cardinal axes, these are the same views. Then, we contend that the one view can be changed into the other by a process of topologically simple distortion. Further, following the conception of Minkowski, our space is locally 'flat'. This is to say that the transformations do not have areas of locally intense distortion but, rather, provide little local change but progressively larger change with distance. By analogy, local space-time is flat, but over cosmological distance, curvature is significant.

This is fortunate because it allows our method of object recognition by matching isolumes to handle universals. As we test for similarity between a gold (sedan) isolume and a brass (SUV) isolume, we see that we are considering five nearest neighbour features. These tend

to be close together and over this short span, the distortion is small. Therefore the angles and distances are not significantly changed and we will tend to record the concordance between the two isolumes and therefore the similarity between the sedan and the SUV. The proportion of isolumes confirmed will decrease as the views become more dissimilar.
We have merely hinted at the way in which the problem of universals can be handled but it seems that the way forward will fall within our compass.

8.4 Flexible Objects: The hand problem

8.4.1 The problem stated

As we consider an image of a hand, composed as it is of, sixteen independently moveable sections, we are struck by how difficult it would be, visually, to decipher the complex structure. This is an extreme example of the general problem of articulated objects (artus L. = joint). This extends by infinitesimal degrees to the problem of flexible objects such as a shark or a hosepipe.

8.4.2 The problem resolved

Mature consideration shows that the joint problem does not fall within the bailiwick of artificial vision because there cannot be any single image of a hand which can show the operation of a joint.
The way in which a joint operates can only be seen from a succession of images and only through the lens of an artificial intelligence apparatus which can make sense of all the constituent parts of the hand and what they are doing in the course of time. How then shall we recognise a static hand? This is an important problem in artificial vision and its gravity is demonstrated by the fact that primates have nerves in the optic bundle which fire only when they see their own hand. Clearly hands spend a lot of time within our field of vision and need to be managed. As a solution, we propose the following;
Consider that the fingers of the hand tend to move in concert as it changes its posture progressively from clenched to splayed. If we declared a clenched fist to be an object, a splayed hand to be another and perhaps two other objects at intermediate conditions, then we could interpolate between these conditions by the direct application of our method. When the hand is adopting strenuous postures such as American Sign Language, we must perforce analyse it as a succession of sixteen objects, each of which is known to us.
The notion of flexible mating between objects (as between the first and second digits of the forefinger) must be grist for the mill of a prepositional calculus – which does not fall within the scope of this chapter.

9. Conclusion

We have outlined a method which, we think, will be adequate for our needs as we proceed to develop an embodied artificial intelligence. The method seems to have no antecedent in the literature and uses concepts which have not been previously considered. We favour the approach of isolume recognition rather than comparison of features or triple matching. It seems that the performance level on our limited dataset is good and that the computational burden is not intractable.
Our work shines a dim light on what might be a broad, sunny upland, rich in promise and new concepts.

10. References

Amores, J.; Sebe, N. & Radeva, P. (2007). Context-based object-class recognition and retrieval by generalized correlograms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 10, pp. 1818-1833.

Ballard, D. & Brown, C. (1982). *Computer Vision*, 1st Edition, Prentice Hall, ISBN 0131653164, Eaglewood Cliffs, New Jersey, USA.

Brown, M. & Lowe, D. (2007). Automatic panoramic image stitching using invariant features, I*nternational Journal of Computer Vision*, Vol. 74, No. 1, pp. 59-73.

Bryson, B. (1990). *The mother tongue: English and how it got that way*, Harper Collins, ISBN 0-380-71543-0, New York, USA.

Conway-Morris, S. (1998). *The crucible of creation*, Oxford University Press, ISBN 0-19-850256-7, Oxford, England.

Da Fontura Costa, L. & Cesar, R. (2009). *Shape Classification and Analysis Theory and Practice*, 2nd Ed., A. Laplante (Editor), Taylor and Francis Group, ISBN 13:978-0-8493-7929-1, Florida, USA.

Dhua, A. & Cutzu, F. (2006). Hierarchical, generic to specific multi-class object recognition, *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 1, pp. 783-788, 0-7695-2521-0/06, Hong Kong, August 2006.

Drew, M.; Lee, T. & Rova, A. (2009). Shape retrieval with eigen-CSS search, *Image and Vision Computing*, Vol. 27, pp. 748-755.

Flemmer, R. (2009). A scheme for an embodied artificial intelligence, Conference Special Session Keynote Address, *Proceedings of the 4th International Conference on Autonomous Robots and Agents (ICARA)*, pp. 1-9, Wellington, New Zealand, February 2009.

Gao, J.; Xie, Z. & Wu, X. (2007). Generic object recognition with regional statistical models and layer joint boosting, *Pattern Recognition Letters*, Vol. 28, pp. 2227-2237.

Gregory, R. (1978). *Eye and brain*, 3rd edition, World University Library, McGraw-Hill, ISBN: 0-07-024665-3, New York, USA.

Hicks, R. (1925). *The lives and opinions of eminent philosophers by Diogenes Laertius*, Translation, W. Heinemann, London, England.

Hutcheson, G. (2005). Moore's Law: the history and economics of an observation that changed the world, *The Electrochemical Society Interface*, Vol. 14, No.1, pp. 17-21.

Hutchinson, J. (2001). Culture, communication and an information age madonna, *IEEE Professional Communication Society Newsletter*, Vol. 45, No. 3, pp. 1-6.

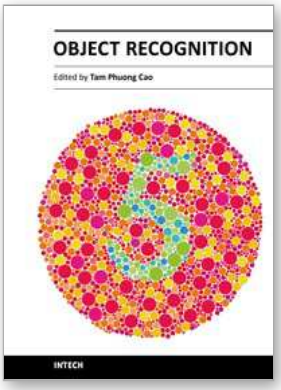
Levi-Setti, R. (1993). *Trilobites*, University of Chicago Press, ISBN 0-226-47451-8, Chicago, Illinois, USA.

Lew, M.; Sebe, N.; Djeraba, C. & Jain, R., (2006). Content-based multimedia information retrieval: state of the art and challenges, *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 2, No. 1, pp. 1-19.

Osher, S. & Fedkiw, R. (2003). *Level set methods and dynamic implicit surfaces*, Antman, S.; Marsden, J. & Sirovich, L. (Editors), Springer, ISBN 978-0-387-95482-0, , New York, USA.

Rosenfeld, A. (1987). *Readings in computer vision: issues, problems, principles and paradigms*, Fischler, M. & Firschein, O. (Editors), Morgan Kauffmann Reading Series, ISBN 0-934613-33-8, pp. 3-12, San Francisco, California, USA.

Valentine, J.; Jablonski, D. & Ersin, D. (1999). Fossils, molecules and embryos: new perspectives on the Cambrian explosion, *Development*, Vol. 126, No. 5, pp 851-859.



Object Recognition

Edited by Dr. Tam Phuong Cao

ISBN 978-953-307-222-7
Hard cover, 350 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

Vision-based object recognition tasks are very familiar in our everyday activities, such as driving our car in the correct lane. We do these tasks effortlessly in real-time. In the last decades, with the advancement of computer technology, researchers and application developers are trying to mimic the humanâ€™s capability of visually recognising. Such capability will allow machine to free human from boring or dangerous jobs.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Rory C. Flemmer, Huub H. C. Bakker and Claire L. Flemmer (2011). Object Recognition using Isolumes, Object Recognition, Dr. Tam Phuong Cao (Ed.), ISBN: 978-953-307-222-7, InTech, Available from: http://www.intechopen.com/books/object-recognition/object-recognition-using-isolumes

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821