

# Autonomous evolutionary machine vision systems.

Jeffrey Johnson and Valerie Rose  
Department of Design and Innovation  
The Open University  
Milton Keynes, MK7 6AA, England  
[j.h.johnson@open.ac.uk](mailto:j.h.johnson@open.ac.uk) and [yrose@ukonline.co.uk](mailto:yrose@ukonline.co.uk)

## Abstract

We propose a radical new approach to machine vision based on biological principles in the context of a multilevel architecture of representation and reconstruction.

## 1. Introduction

Machine vision is a notorious bottleneck in robotics and automated systems. We seek a method of creating very flexible machine vision systems that can evolve in particular environments to recognise anything that an operator has indicated as being ‘interesting’ in that environment. For example, Figure 1 shows an object that a house-tidying robot might encounter during its everyday duties.



**Figure 1.**Contouring an object

Our intention is that non-programmers can train our vision systems by ‘pointing’ at an object in a scene, *e.g.* drawing a contour round it, with the system to evolving the ability to recognise such objects automatically. Our approach is based on new combinatorial structures supporting an *architecture* that allows vision systems to generalise and adapt to recognise new classes of objects. This architecture is based on new combinatorial mathematics in the science of complex systems [1].

There are many approaches to machine vision, including algorithmic knowledge-based programming, neural systems that learn from examples, and combinations of both. Many practical systems are based on algorithms or procedures making opportunistic use of special features of particular objects and scenes. Although this may give acceptable performance for a given application, there is usually a poor ability to *generalise* to other similar scenes, and no ability to generalise to different environments. A system designed to inspect industrial parts is unlikely to be incorporated in a mobile planetary robot.

It is common for human programmers to *design* vision systems so that data are optimised for the particular problem and classification technique being used. The generality is that machine vision systems are hand-crafted to give the best results for a particular application, but are brittle and perform poorly outside their narrow specification, and lack any ability to adapt.

We seek machine vision systems that can:

- use point-and-learn training
- work for cluttered scenes
- adapt to changes in objects and scenes
- adapt to any scene or environment

To achieve this we propose a multilevel architecture in which machine vision systems

- evolve appropriate retinal configurations
- evolve connectivities to represent spatial relationships
- abstract their own higher level constructs
- levels are integrated by new relational mathematics

The key feature of the architecture is the ability of the system to abstract its own constructs from data in a multilevel algebraic representation. This allows the system to learn objects that may change through time, and to

adapt to learn radically new objects and scenes without the need to change the underlying program. These requirements are very demanding and beyond any existing machine vision systems.

## 2. The Fundamental Structures

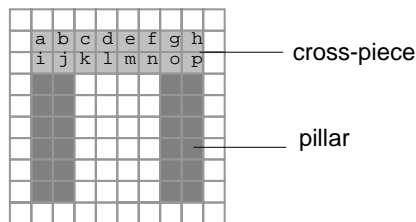


Figure 2. An image of an arch.

The fundamental idea behind our architecture is that of *n-ary relations*. To illustrate this consider the image of an arch in Figure 1. As we view it, we see two *pillars* supporting a *crosspiece*. The crosspiece, for example, is made up of the pixels marked a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, and p. These sixteen pixels are *assembled* by a 16-ary relation,  $R_{RB}$ , into a rectangular block.

The *construct* ‘rectangular block’ defines a set of objects which will be written  $RB = \{x \mid x \text{ is a rectangular block}\}$ . In order to be operational, this requires a *pattern recogniser*,  $P_{RB}$ , which is able to say of any candidate for membership,  $x$ , that  $x$  is a rectangular block,  $P_{RB}(x) = \text{True}$ , or that  $x$  is not a rectangular block,  $P_{RB}(x) = \text{False}$ .

Generally pattern recognisers need to refer to a set of features of the object. In this case there are sixteen features,  $\{x_1, x_2, \dots, x_{16}\}$ , the pixels used to make up the block. Each of these  $x_i$  needs to be of the right type, so the overall pattern recogniser requires a set of sub-pattern recognisers,  $P_{RB,i}$ , with the requirement that  $P_{RB,i}(x_i) = \text{True}$ .

Now it can be seen that the pattern recognition involves two types of decision:

- (i) for each  $x_i$ , it is necessary that  $x_i$  is of the right type, here a dark pixel.  $P_{RB,i}(x_i) = \text{True}$ .
- (ii) given that all the  $x_i$  are pixels, it is necessary to established that they are assembled properly so that the relational structure holds with  $P_{RB}(x_1, x_2, \dots, x_{16}) = \text{True}$ .

Clearly (i) comes before (ii). There is no point applying expensive pattern recognition procedures to objects which are of the wrong type to form the pattern. However, there is

danger of an infinitive regress: To test  $R_{RB}$  it is necessary to test  $R_{RB,i}$  for each  $x_i$ . But to test  $R_{RB,i}$  it is necessary to reduce  $x_i$  to its parts, and test them. And so on. Where can it all end? In robotics and machine vision the answer to this question is easy. Top-down reductionism ends when the pattern recognisers are *grounded* in sensor data. In other words the sensors ‘ground’ everything the machine can know about its environment (Figure 3).

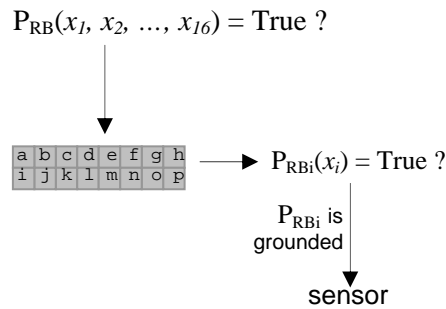


Figure 3. Reductionist grounding prevents infinite regress in pattern recognition

When  $P_{RB}(x_1, x_2, \dots, x_{16}) = \text{True}$  for a particular set of features,  $\{a, b, c, \dots\}$ , we give the resulting object a name, here  $\mathbf{C}$ , and write  $\sigma(\mathbf{C}) = \langle a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p; R_{RB} \rangle$ .  $\sigma(\mathbf{C})$  is called a *simplex* and the elements  $\langle a \rangle, \langle b \rangle, \langle c \rangle$ , etc are called its *vertices*. The parts or features of an object can be said to lie at a lower level in its representation than the object itself. If the parts are drawn within an Euler circle (ellipse), the name of the object can be drawn as the apex of a *hierarchical cone*, as illustrated in Figure 4.

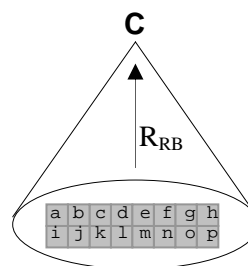
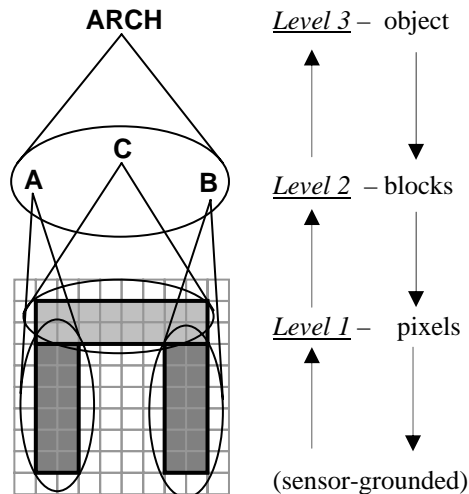


Figure 4. A hierarchical cone.

The relation  $R_{RB}$  and all its reductionist sub-relations will be called a *construct*. Clearly, in order for a construct to be operational, it must be grounded. Generally constructs are *named*, and they define sets of named objects. E.g., we can use the name ‘rectangular blocks’, and write rectangular blocks =  $\{x \mid x \text{ is a rectangular block}\}$ , which is an intensional definition. Alternatively we can write Rectangular Blocks =  $\{RB_1, RB_2, \dots, RB_n\}$ ,

where each of  $RB_i$  is the name of a particular rectangular block.

Figure 5 shows the two stages of assembly of the arch; which is defined as structured set of blocks. The blocks are defined as structured sets of pixels; and the pixels are grounded in reality through the camera sensor.



**Figure 5. Multilevel construct aggregation**

The pillars named as A and B in the image are also structured sets of pixels, as shown in Figure 5. The intermediate constructs A, B and C can be assembled by a 3-ary relation,  $R_{arch}$  to form the construct called an ARCH. Thus we have  $\sigma(\text{ARCH}) = \langle A, B, C; R_{arch} \rangle$ . In this way we build primitive structures from *atomic constructs* (pixels), we build *intermediate constructs* from these at a higher level in the multilevel representation, and so on, until we recognise objects within scenes at the highest level. At every level we use named constructs to reference the objects abstracted.

This example illustrates a major problem in machine vision. The notions of ‘pillars’ and ‘crosspieces’ are social constructs inside programmers’ heads. Vision systems are highly dependent on programmers’ ways of construing the visual universe. It is well known that this can be very different between different people [2], and there is no guarantee that a given programmer will choose the most appropriate constructs. Much better to have the vision system abstract these constructs for itself.

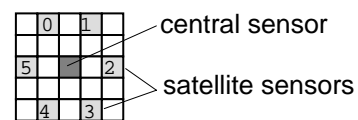
### 3. Low level pixel configurations

In the proposed multilevel architecture, let the pixels define a base level, *Level 1*. (Lower level sub-pixel constructs are possible, (e.g. Johnson and Picton, 1985), but not discussed

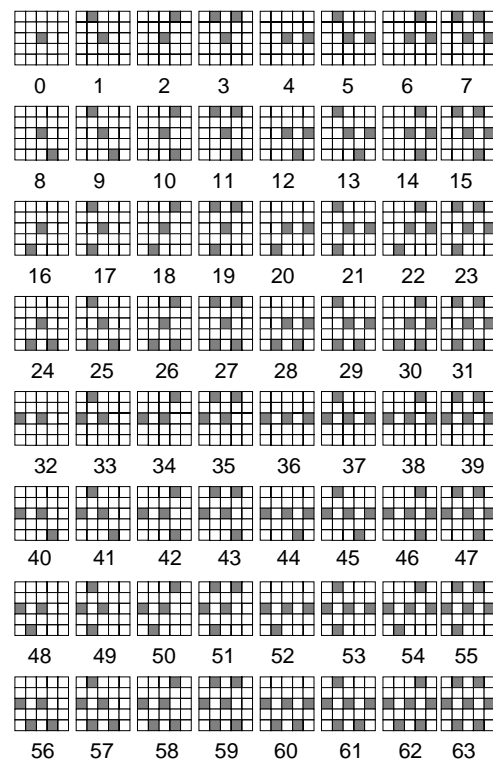
here. At this level of representation are the usual greyscale histograms.

The next level of representation must be characterised by sets of pixels structured by relations – nothing else is possible! So, *Level 2* in the representation will consist of sets of pixels under *n*-ary relations. To illustrate this, consider the pixel configurations shown in Figure 6. To establish them at the lowest level in the representation, these will be called *retinal constructs*.

In Figure 6(a) there is a central sensor, such as light-sensitive rod, responding to relative darkness, surrounded by six other sensors, numbered 0, 1, 2, 3, 4 and 5. There are  $2^6 = 64$  configurations of light-dark for these six *satellite* sensors. The configurations have been designed to have a topology corresponding more closely to packed cells than the usual Cartesian grid. Also they are designed using the ‘next but one’ *neighbours* according to Simon’s three pixel principle [3][4].



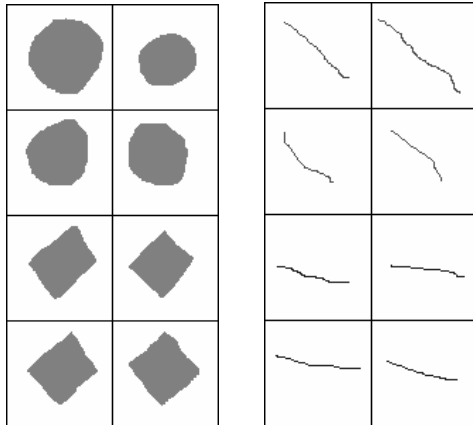
(a) hexagonal array of pixel sensors



(b) the 64 retinal configurations

**Figure 6. Hexagonal pixel constructs**

These configurations are examples of *masks* or *filters* which are widely used in machine vision. As such they have been *designed* by the programmer (me!) and have the problem of subjective selectivity. Although I find these configurations attractive for a number of reasons, how can I be sure that they are the most appropriate for any particular objects in any particular environment?



(a) circle and diamonds (b) line segments

**Figure 7. Examples object classes**

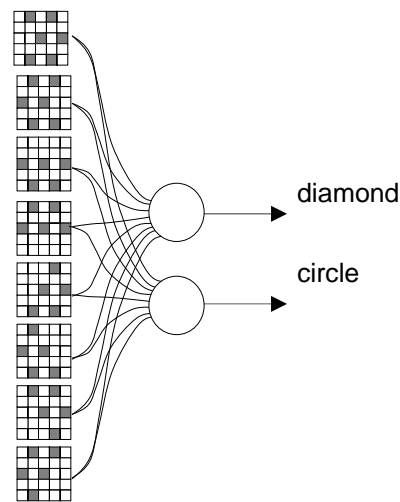
The sixty four retinal configurations in Figure 6 were used to analyse eighty hand-drawn shapes, forty 'circles' and forty 'diamonds', similar to those shown in Figure 7. Each dark pixel in the shapes was analysed by inspecting its surrounding pixels and assigning to it one of the sixty four retinal configurations. As a first level of analysis, the numbers of each configuration were counted, giving a 64-element vector for each configuration. The vectors of the configurations with non-zero frequencies are given in Table 1.

#### 4. Single Level Neural Classification

Inspection of Table 1 suggests that the frequency vectors alone are sufficient for classification of the simple circle and diamond shapes, and indeed they are. For example, configurations 14 and 31 have much higher

frequency for the circles than the diamonds, reflecting their natural response to vertical left and right edges respectively. Similarly, configurations 7, 30, 51 and 57 favour the diamond shape by responding well to oblique edges.

In principle, a conventional multilayer perceptron neural network will classify such data well, assuming convergence. Note that in Table 1, twenty nine of the sixty four possible configurations respond to the eighty shapes, leaving thirty five retinal configurations that do not respond to these shapes. Training the network with all sixty four configurations as inputs increases the computation and the possibility of non-convergence.



**Figure 5. A single level vector neural classifier**

Table 2 gives the configuration counts for the line segments shown in Figure 4(b). The response of these objects to the retinal configurations is completely different to that for the shapes. These response vectors can also be used for robust classification between the steep and shallow line segments. It is encouraging that a single layer neural classifier can discriminate these line segments, since it is believed that animal vision uses such primitives.

|         |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
|---------|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| diamond |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 3       | 4 | 6 | 7  | 12 | 14 | 15 | 24 | 28 | 30 | 31 | 32 | 33 | 35 | 39 | 46 | 47 | 48 | 49 | 51 | 53 | 55 | 56 | 57 | 59 | 60 | 61 | 62 | 63   |
| 2       | 1 | 1 | 22 | 1  | 1  | 21 | 2  | 25 | 24 | 0  | 1  | 2  | 21 | 0  | 1  | 21 | 1  | 2  | 25 | 1  | 21 | 18 | 24 | 2  | 0  | 17 | 24 | 978  |
| 2       | 1 | 1 | 25 | 1  | 2  | 21 | 1  | 14 | 26 | 1  | 0  | 1  | 18 | 1  | 1  | 24 | 1  | 2  | 18 | 0  | 17 | 25 | 27 | 0  | 2  | 23 | 12 | 885  |
| 1       | 0 | 1 | 27 | 2  | 4  | 23 | 0  | 27 | 17 | 3  | 0  | 0  | 25 | 2  | 0  | 25 | 2  | 2  | 28 | 0  | 22 | 28 | 19 | 0  | 4  | 26 | 25 | 1256 |
| 2       | 0 | 0 | 30 | 1  | 3  | 17 | 2  | 26 | 29 | 0  | 0  | 0  | 22 | 5  | 0  | 28 | 1  | 3  | 28 | 0  | 20 | 21 | 31 | 0  | 3  | 20 | 25 | 1292 |
| circle  |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 3       | 4 | 6 | 7  | 12 | 14 | 15 | 24 | 28 | 30 | 31 | 32 | 33 | 35 | 39 | 46 | 47 | 48 | 49 | 51 | 53 | 55 | 56 | 57 | 59 | 60 | 61 | 62 | 63   |
| 0       | 0 | 2 | 14 | 0  | 8  | 20 | 0  | 25 | 19 | 6  | 0  | 1  | 11 | 31 | 0  | 12 | 0  | 22 | 17 | 0  | 8  | 10 | 14 | 19 | 21 | 6  | 21 | 1322 |
| 0       | 0 | 0 | 18 | 2  | 18 | 10 | 0  | 6  | 18 | 16 | 0  | 0  | 8  | 33 | 0  | 14 | 2  | 16 | 18 | 0  | 4  | 17 | 13 | 14 | 32 | 15 | 4  | 1253 |
| 0       | 0 | 2 | 13 | 1  | 11 | 37 | 0  | 10 | 12 | 10 | 0  | 0  | 12 | 28 | 0  | 11 | 1  | 14 | 16 | 0  | 8  | 27 | 15 | 11 | 23 | 24 | 7  | 1375 |
| 0       | 0 | 1 | 18 | 1  | 14 | 10 | 2  | 14 | 18 | 12 | 0  | 1  | 8  | 23 | 0  | 15 | 1  | 22 | 14 | 0  | 5  | 10 | 12 | 20 | 19 | 9  | 13 | 1083 |

**Table 1. Frequencies of retinal configurations in the shapes of Figure 4.**

However, these classifier soon break down when the number of objects to be classified gets large, as is required for recognising a comprehensive set of line segments.

| steep lines |   |    |   |   |   |    |    |    |  |
|-------------|---|----|---|---|---|----|----|----|--|
| 0           | 1 | 4  | 5 | 8 | 9 | 32 | 36 | 40 |  |
| 9           | 0 | 13 | 2 | 0 | 0 | 13 | 21 | 2  |  |
| 3           | 0 | 11 | 1 | 1 | 0 | 12 | 27 | 0  |  |
| 8           | 1 | 17 | 1 | 0 | 0 | 16 | 15 | 2  |  |
| 7           | 0 | 11 | 0 | 0 | 0 | 11 | 28 | 0  |  |

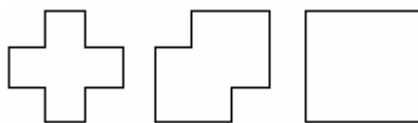
| shallow lines |    |   |   |    |    |    |    |    |  |
|---------------|----|---|---|----|----|----|----|----|--|
| 0             | 1  | 4 | 5 | 8  | 9  | 32 | 36 | 40 |  |
| 21            | 7  | 5 | 1 | 7  | 4  | 5  | 1  | 1  |  |
| 32            | 4  | 0 | 0 | 4  | 5  | 0  | 0  | 0  |  |
| 30            | 10 | 1 | 1 | 10 | 18 | 1  | 0  | 1  |  |
| 28            | 11 | 1 | 3 | 13 | 10 | 3  | 0  | 1  |  |

**Table 2. Line segments frequencies ( Fig 4)**

The approach to pattern recognition illustrated here map the object to a vector of numbers counting the frequency of ‘interesting’ features of the objects, interprets the vector as a point in multidimensional space, and classifies the points according to some notion of ‘similarity’. In terms of our objects it begs two questions:

1. where do the ‘interesting’ features come from?
2. is a single level of processing adequate to discriminate objects in complex scenes?

In answer to first question, in our illustrative application, the ‘interesting’ features were designed in by the programmer. Delegating the selection of ‘interesting’ features to a programmer inevitably means that the system will be limited in its ability to recognise objects, and unable to adapt to recognise objects that are very different from the design specification.




**Figure 9. Shapes with equivalent vertical and horizontal lengths**


It is easy to show that this kind of single level of classification is inadequate in general for object recognition in vision. For example, the objects in Figure 9 all have the same length of vertical and horizontal edges. Conceivably the corners would have different retinal configurations, but the numbers would be small, and robust discrimination between the objects by a single vector of retinal configurations is unlikely.

The answer to the second question must be that a single level of classification is not adequate. If it were, objects and scenes could be presented to a network as an input vector, to deliver recognition of classified objects. Even if this were possible in theory, it would be impractical because combinatorial explosion mean that the necessary input vectors would have astronomic numbers of elements.

## 5. Interpreting the data as constructs

In the previous section it has been seen how some retinal configurations can be associated with constructs such as ‘oblique’, ‘vertical’, ‘left and ‘right’ edges. These are human constructs that can be imposed on the data. The machine, of course, does not share these constructs explicitly in its representation. Thus there is a co-relation between our concept of a ‘round edge’ and, say, the pixel configuration 49, , taking a relatively high value for circular objects.

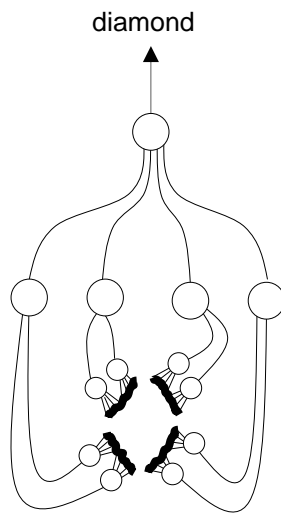
Put like this it becomes possible to understand why conventional approaches to machine vision have failed so comprehensively. As programmers we seek appropriate descriptors or constructs to represents objects to be recognised. We look at an object, and abstract properties such as ‘roundness’ and ‘straightness’, that our language conveniently has terms to describe. We then seek machine-based abstractions that match these linguistic constructs.

But as animals we constantly recognise objects for which there is no explicit name. For example, most readers will recognise the shape  as being one of those in Figure 9, even though this shape has no explicit common name. Since I want to talk about it I will give the name of ‘double-square shape’. Then I can say things like ‘the double square shape is between the cross and the square in Figure 9, and even begin to reason about double-square shapes. However, if such a shape were to be recognised within a machine, it can simply be named implicitly by the data structures, possibly, its position in memory.

Freeing ourselves from serendipity abstractions in a particular programmer’s head, and designing machines to form their own constructs is here seen as the way forward. To some extent this is what multilayer neural networks do, and some researchers assert that each neuron is processing a construct. However, that approach is relatively blunt, since the constructs are always implicit.



relationship between them is established (and computed) by located connections between the site of response and higher level processing that recognises the object.

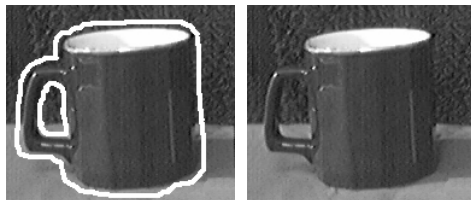


**Figure 12. Hard-wired spatial relationships**

## 7. Segmentation

One of the most difficult tasks in machine vision is to segment a complex scene into 'relevant' parts. Generally one seeks areas that contain discrete objects, such as the coffee mug shown in Figure 10. In Figure 10(a) we show a training object identified by a user. This low-skilled method of teaching the system is the only kind of input the trainer gives. Following this, the system has to find discrete objects in the image to be recognised as of the same type as the training items.

Figure 10(b) illustrates the many problems involved in segmenting images. The mug has no well defined contour, since neither it nor the background are homogeneous in greyscale. In some places the mug merges into the dark background, while in other places it is a relatively light grey due to the reflected light.



**(a) user defined object (b) how to segment?**

**Figure 10. The segmentation problem**

The white ellipse is a strong signal to humans that this is a cylindrical object, but the machine

knows nothing of this *a priori*. In many places the mug has highlights, making it visually very variable. In the first instance we do not assume that that our system will have top-down context knowledge such as 'if the scene contains an ellipse, then it contains a cylinder'.

Although the examples in this paper have been pre-segmented binary images, the methods developed here are highly applicable to greyscale and coloured images. The relational method can be very powerful when, for example, the satellite pixels are compared to the centre and classified as light/darker. This approach can lead to areas with varying greyscales but homogeneous greyscale gradient. This approach has been successfully applied in scientific measurement systems. The details are beyond the scope of this paper, but further details can be found in [3][4].

## 8. The architecture

The research described in this paper, in the context of our objectives leads us to the following principles:

Principle 1. Low retinal configurations will aggregate to form higher level constructs

Principle 2. The constructs will depend on spatial relations

Principle 3. The retinal configurations should not be constrained by design, but should be allowed to emerge from the images and scenes in its environment

Principle 4. The spatial relations in the system should be implicit in its topology so that Cartesian geometry need not be used

Principle 5. Higher level spatial configurations should not be constrained by design, but should be allowed to emerge from the images and scenes in its environment

Principles 1 and 2 are the fundamental theoretical underpinning of our approach. They are supported by algebraic mathematics that can be implemented as data structures in real computers.

Principle 3 is based on the need for the system to adapt to new things. Any system with pre-designed primitives is constrained by what the designer puts in. This *spans* the space of possibilities. Any object not in that space cannot be recognised. This is one reason why

conventional machine vision collapses outside its design domain.

If the low level configurations are not to be designed in, where can they come from? We have experimented with forming low level constructs by random configurations of pixels. We have found that generating random masks gives some discrimination between the circular and diamond shapes discussed earlier. However, there remain many open questions, including the optimum diameter for a retinal configuration.

A similar argument suggests there can be no fixed multilevel architecture, and this too must incorporate random processes. Thus the 'relevant' configurations of configurations, and the resulting 'construct' have to be discovered by the machine.

Thus there are two parts to our architecture. The simplest involves the machine learning particular objects and scenes within a given hardware topology. In other words, in the simplest case the machine is fixed, and recognition takes place by values and parameters changing within that structure.

The more demanding part of the architecture involves evolutionary principles to generate and select 'appropriate' retinal primitives, and to generate and select appropriate topologies to support relational structure throughout the multilayer aggregation. This lies at the heart of our research programme to achieve autonomous evolutionary machine vision systems.

## 9. Discussion and conclusions

In this paper we have illustrated some of the basic ideas of our research programme, and reported briefly some preliminary experiments on evolving retinal configurations. Those experiments combined with the experiments reported on designed retinal configurations suggest that this part of the research will be relatively straight forward. In other words, the research on the evolution of retinal configurations has already begun and we are beginning to understand this part of the challenge relatively well.

By far the greatest challenge is in implementing the multilayer architecture to support the hierarchical assembly of information towards object and scene recognition. One major unresolved challenge in this is to design spatial structure into the

system in a way that overcomes combinatorial explosion. The human brain has some ten billion neurons with five to ten thousand connections per neuron. This apparently huge amount of processing power and information distribution ability appears more modest when compared to the numbers of ways that retinal configurations can be defined and connected in a multilevel architecture. Furthermore, we expect to implement our architecture on standard computers with orders of magnitude less memory and orders of magnitude less computational ability than biological vision systems.

Currently our idea is that spatial structure is determined by the initial connections to the retinal configurations, where the image is grounded, and subsequent connections are through the multilayer system. There is a major challenge in establishing a theoretical architecture that can be implemented in practice, followed by the major practical challenge of inducing the system to self-organise as it adapts to new visual environments.

We are aware of the difficulty of the research we propose. We are optimistic that success is possible because the combinatorial mathematics underlying our research contains some of the essential structures necessary to achieve our objectives.

## References

- [1] Johnson, J.H., 'Some structures and notation of Q-analysis', *Environment and Planning B*, **8**, 73-86, 1981.
- [2] Gregory, R. L., *Eye and brain: the psychology of seeing*, Oxford University Press, 1998.
- [3] Johnson, J.H., Picton, P.D., *Mechatronics: Concepts of Artificial Intelligence*, Butterworths (London), 1985
- [4] Johnson, J.H., Simon, J-C, 'Fundamental Structures for the Design of Machine Vision Systems', *Mathematical Geology*, Vol. **33**, No.3, 2001.
- [5] Marr, D., *Vision*, W Freeman and Company, (New York), 1982.
- [6] Rose, V., Results of experiments in shape-recognition through random selection of sets of pairs of pixels within 75x75 pixel binary images of various simple shapes. Mimeo Open University, 2004.