Non-object based visual navigation.

D.L.Bisset M.Aldred
Electronic Engineering. Laboratory
University of Kent at Canterbury
Canterbury
Kent
CT2 7NT
D.L.Bisset@ukc.ac.uk & mda1@ukc.ac.uk

Introduction

Navigation needs to be achieved at many different scales. For this reason it represents a "rich" problem for autonomous robotics research. Navigation tasks can typically be divided into three levels each at a different scale of the environment:

- local navigation to avoid obstacles,
- room and corridor level navigation,
- building level navigation.

At each scale different strategies can be used to provide solutions. In such a hierarchic system each strategy relies on the one below providing correct responses to the sequence of sensory data in order for it to work correctly.

In most cases fixed methodologies are used at each level. These methodologies often make use of the different types of structure at each level so, for example, building level navigation is most often carried out on an accurate Cartesian map, room and corridor level navigation is done in terms of wall or corner features(e.g. [Mataric]), and at the local level reactive systems are used(e.g. [Brooks]).

Each of these methods requires a certain amount of prior knowledge about the environment. The Cartesian case requires detailed measurements to be taken, these can be extracted from building plans or by "walking" the route with the robot entering the location of each room or corridor junction as it is reached. The feature maps can be constructed in the same way, but rely on prior knowledge of the features that could be encountered (e.g. floors, walls, corners in an office environment), so that they can be coded into a map structure. Although the particular features encountered can be described independently of the sensing system they are still bound to a particular environment, for example an ultrasonic sensing system might recognise the feature generated by an infinite distance reading on the left (as well as front and back) as being a left corridor, and a similar structure might also be recognised by a vision system able to detect a carpet texture to the left, where previously there had been a wall texture. Even the reactive approach relies on embedded knowledge about the environment, this knowledge (e.g. "proximity is undesirable") is embedded in the type of sensors used and the controller structure where, typically, distance to object or goal is used to derive a force vector that guides the vehicle. In each of these different approaches the critical issue is the level of generalisation that can be achieved.

Unstructured environments test the level of generalisation of a particular navigation method. The success of reactive based schemes comes from their reliance on the extraction of non-task specific parameters and non-object specific parameters from the environment. This makes the resulting action applicable, within the limitations of the sensors, to any environment. On the other hand a Cartesian map solution is only applicable to environments where Cartesian maps can be reliably constructed and referenced to the real world. A particular instance of a Cartesian map solution is only applicable to the environment where that particular map is valid, this type of solution therefore has a low level of generalisation.

An additional problem for these approaches is the task of referencing the stored knowledge to the sense data. For the Cartesian map it may be possible to use a positioning system such as GPS, for systems based on features no easy solution is available. It is notable that human road maps are feature based with respect to the objects that can be encountered, such as a bridge or telephone box, simply because we do not possess accurate "distance travelled" sensors, and prefer to rely on our more sophisticated feature location and identification abilities.

The problem of referencing the stored features or map to the world, or the robot to the map varies in difficulty depending on the environment and on the data in the map. Successful implementations of navigation systems based on Cartesian building maps rely on accurate motion assessment, and error correction. Successful feature based maps rely on unambiguous feature recognition and location [Mataric] [Nhemzow].

The level of specific knowledge about unknown and unstructured environments is by definition limited, however every environment obeys the basic laws of physics, typically mechanics and optics are the most important, as do the dynamics and sensing systems of any particular robot, (although gross changes such as moving from land to sea cause very significant changes). In order to navigate in unstructured and unknown environments and improve the level of generalisation of the method used it is important to base the system of navigation on constant environmental factors that are transferable between situations.

Vision Systems

Vision systems have not been the sensor of choice for mobile robot navigation because of their high processor overhead, and the lack of a general purpose vision framework for dealing with unstructured images. Many special purpose vision systems have been constructed, but these typically either lack generalisation by concentrating on line extraction and reconstruction, or concentrate on a single environmental feature such as colour or texture.

There has also been a tendency to rely on a small subset of computer vision techniques which typically do not scale well, or are difficult to manage. For example most stereo vision systems require careful calibration, and will provide poor information if the orientations of the cameras are disturbed, as might occur moving over rough ground [Horswill]. This has lead to the use of laser range finders to measure distance, based on the fact that laser range finders are better distance measuring devices than stereo camera pairs, however this ignores a more basic question of whether or not *distance measuring* is what is required to do navigation. [Smithers].

Vision systems based on pattern recognition suffer from a number of problems. Firstly example patterns must be identified to represent environmental features, objects or locations. This frequently leads to a "blocks world" approach where the environment has to be heavily engineered to enable the vision system to work thus compromising the general applicability of the algorithms used. Alternatively it leads to crude scene matching based on pixel difference measures [Horswill]. In general, pattern recognition methods rely heavily on situation specific prior knowledge, either in terms of example patterns or in terms of the manipulation of image processing primitives specifically designed for the environment in question, for example extracting the edges of roads in order to drive between them.

Self-organising neural network techniques such as the use of Kohonen maps [Kohonen] [Nehmzow] provide a means of applying a general purpose algorithm to each specific environment. In general it is usually difficult to extract general purpose reusable knowledge once training is complete, and training is often a long and unsuccessful process. In an unstructured environment the goal must be to extract and store as much useful information about the environment at each exposure, and to do so in such a way that the knowledge is reusable in different environments. In addition Kohonen maps suffer from a severe limitation, that the granularity of the map is fixed when it is constructed. This limits the number of different features that can be detected, and limits the possibility of extending the map after it has been exposed to data since the information is distributed over the weights, in addition any new weights added will reduce the response to already learned stimuli.

Any navigation system relying on vision that can operate in unstructured environments must be able to fulfil the following criteria:

- Be able to learn quickly, but in a flexible way that allows the addition of new knowledge on a continuous basis.
- Not be designed for any specific environment.
- To be able to generalise its stored knowledge to new situations.

Rao and Ballard's AVA

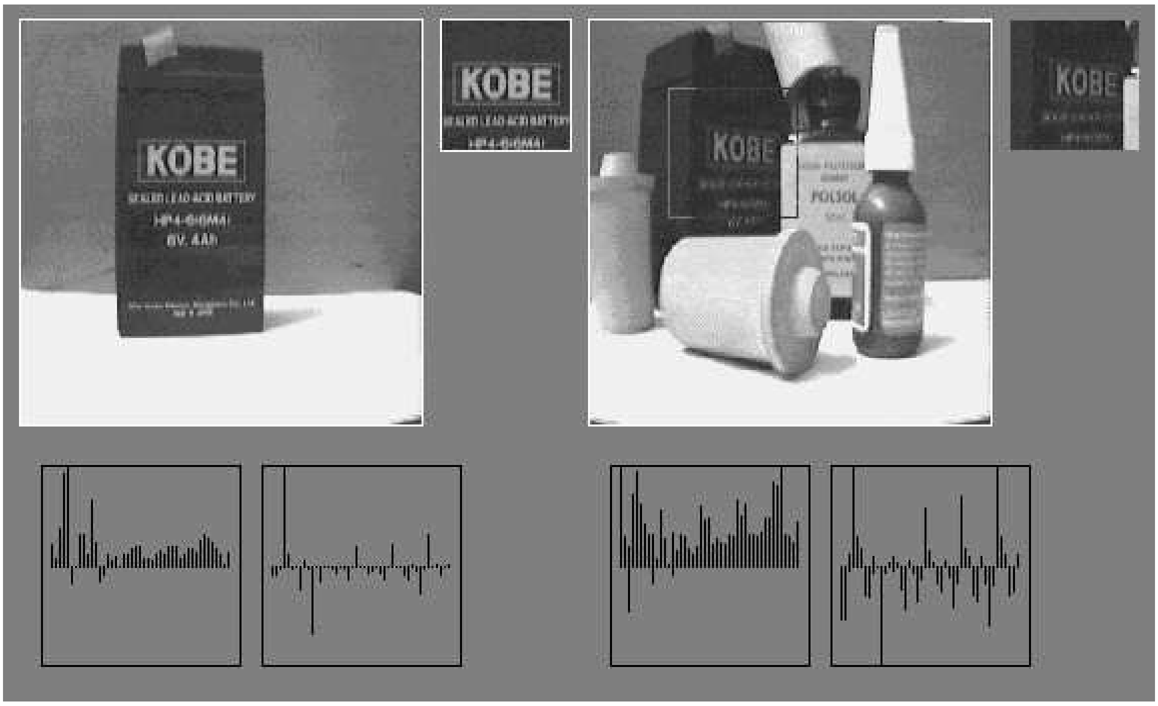
Rao and Ballard have described a general purpose vision system called AVA [Rao]. This is a non-object based vision system that extracts a 45 element vector of information, known as a ZIP code, for any selected pixel patch in the image, the system is then able to scan a similar scene and locate the corresponding patch. The system uses a multi-scale approach coupled to the extraction of rotation invariant features.

The system is independent of any particular object characteristics and is therefore ideal for use as the basic framework of a vision system for mobile robot navigation in unstructured environments.

In this paper the AVA is used to provide the first stage of the image processing system used for navigation. Interestingly the feature primitives that make up the 45 element vector are approximations to the principle components of general images, and are similar to the self-organising cells found by [von der Malsburg] and [Linsker].

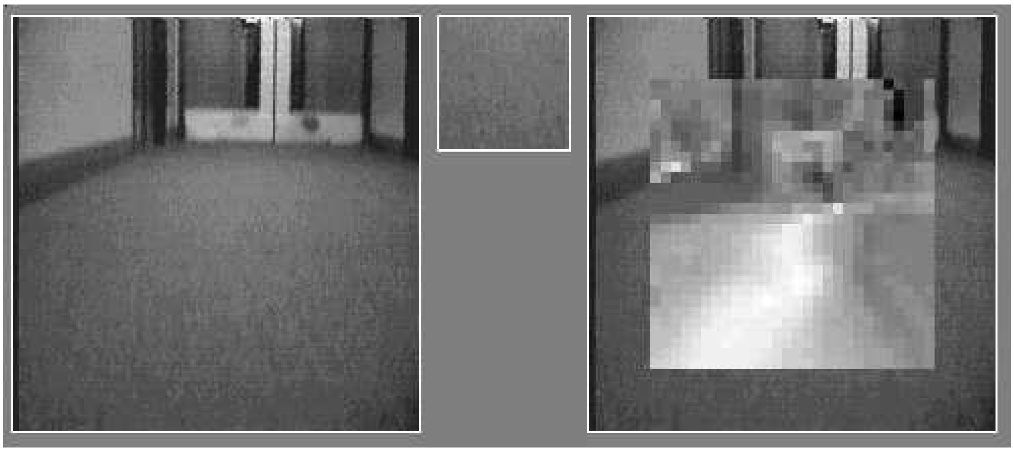
Using the AVA for navigation

The AVA is based on the generation of a measure of similarity between a particular pixel patch in a scene and any other pixel patch in that or any other scene. Figure 1 shows this for a simple image where a pixel patch has been selected from the left hand frame, (the patch is shown in the small picture to the left) and the right frame shows the location of the most similar patch that has been located in the test image. The test is performed on an image where the object has been rotated 30 degrees in the plane, and other objects have been inserted into the scene.



• Figure 1: AVA system patches located on two different images together with ZIP codes.

For visually guided navigation a means of detecting similar visual structures is required. If a pixel patch is selected from the floor in front of a robot then it can be used to match the rest of the floor in the current image. Provided the computation speed is high enough then an area of floor space in front of the robot can be continuously scanned to identify usable floor area as the robot proceeds through its environment. Figure 2 shows this process. In this figure the left hand frame shows a test image, and the



• Figure 2: Corridor floor detection

right hand frame shows a plot of intensity values that correspond to the match between the test patch and the image. The bright pixels indicate points of close match. The boarder around the image is left to avoid boundary effects, and the testing is done on a five pixel step horizontally and vertically across the image. Clearly the algorithm has spotted the floor area similar to the test patch in the image.

A control algorithm that makes the robot move onto a pixel patch similar to the one it is currently on will force the robot to drive through the corridor, updating the current patch as it goes. This strategy will fail if the scene changes suddenly, say as a result of the sun coming out to provide a bright patch on the floor, in which case the robot may attempt to avoid this area, or if the robot, due to a navigation failure, comes to be on a dangerous surface, such as a staircase, the algorithm will then tend to keep the robot on that surface. Figure 3 shows the response to a pre-stored floor patch in a more complex scene. In this image the bright path in front of the robot clearly diverges round the central obstacle, however the band of light across the corridor causes a block, as does the staircase edge.

These examples clearly indicate the potential that the AVA system has for detecting useful features of

• Figure 3: Pre-stored floor patch applied to a different scene.

the environment. In this test the floor patch has been selected in advance, for a robot working in an unstructured environment it will be important for it to recognise various dangerous situations in advance of encountering them, and without them being pre-labelled. The navigation sensing system must therefore extract and organise its own patches.

Any approach to visual navigation must attempt to recognise similar patches and automatically associate them with motor actions. One possible technique might be to employ a Kohonen map connected to the ZIP code vectors. The problem with this is that the map must see examples of each possible input pattern within a fixed number of exposures to the image data otherwise suitable feature detecting cells, locations on the map, will fail to develop. Once the map is trained it becomes difficult to add new cells, because the information storage is distributed and new weights will weaken the existing feature detection mechanism. This means that the system will be effectively frozen after a fixed time, or after a certain number of features have been stored/detected, whether or not it has experienced a broad range of images.

These problems are common to many neural network solutions which rely on a binary or threshold based coding at their output. A solution to these problems can be found in the use of networks that utilise the output nodes of a network to define a flexible feature space that can adapt without necessarily loosing already learnt feature detection properties. Principal Component Networks as described by [Plumbley] provide a partial solution to these problems. These networks rely on a mixture of Hebbian and Anti-Hebbian learning to extract the N principle components from the input data. Although there are a number of different methods for calculating the principle components of a data set the use of PCN provides a computationally efficient method of providing a continuous convergence towards the principal components of an infinite data set. PCN being based on neural network technology also allows other neural techniques to be smoothly integrated with the proposed system.

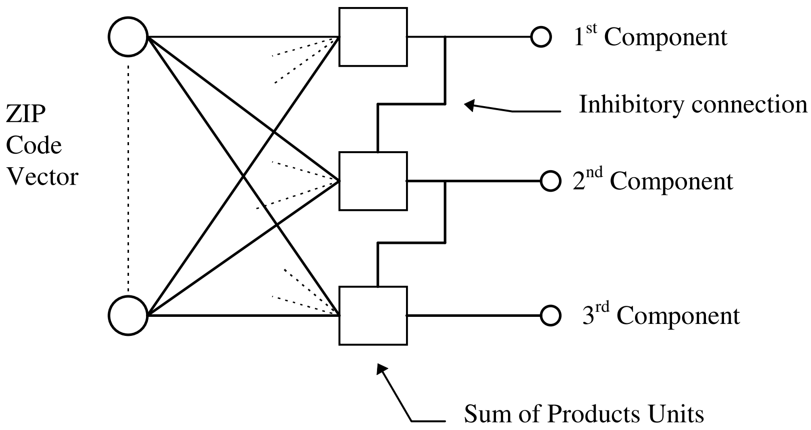


Figure 4: A Principal Component Network.

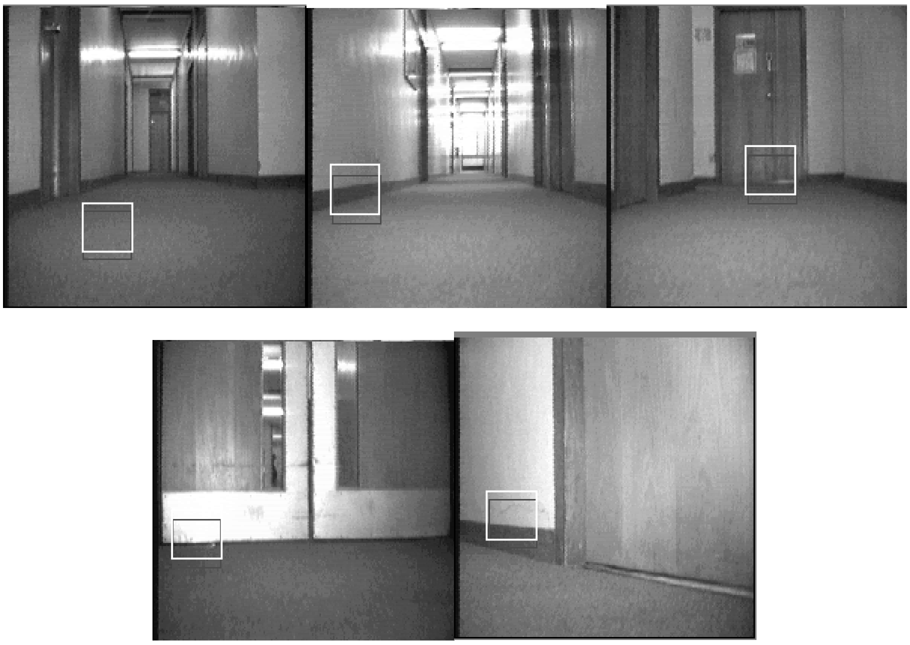
Figure 4 shows the PCN structure used in this paper to extract the first three principle components from the ZIP codes. It is important to note that this processing structure is not dedicated to any particular type of scene or environment. The following sections detail the results of using this PCN structure on the corridor scenes.

Results from Principle Component Networks

A series of different colour images of corridors was taken from around the Electronic Engineering Laboratory under different lighting conditions. These images included junctions, doors and long stretches of corridor. From these images 500 sample patches from particular points in the images were extracted and labelled as follows

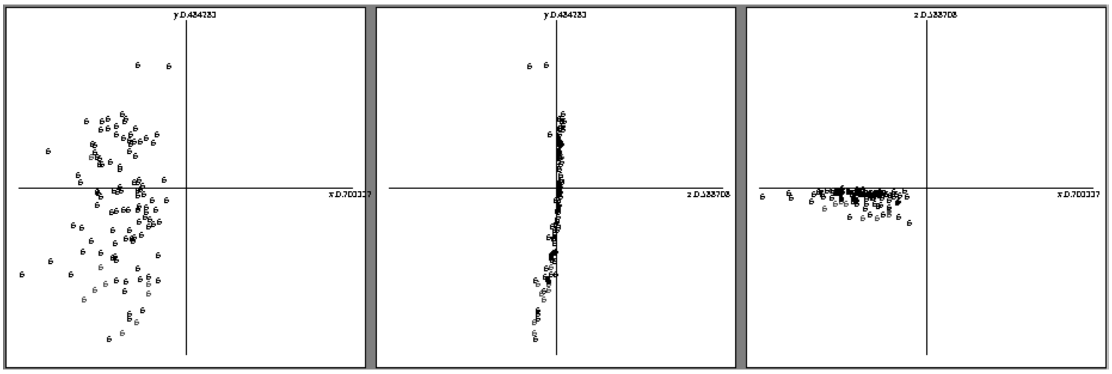
- Floor patches.
- Skirting board transitions from floor to board and from board to wall on both left and right hand sides of the corridor.
- Metal door kick panels.
- Wooden doors.

Figure 5 shows a sample of the patches and scenes used. The ZIP codes used in the following experiments differ from those used by Rao and Ballard in that the two smaller patch sizes have been removed (2 by 2 and 4 by 4 pixel operators) this is because these two components have a very high variance in the images in this data set and as a result they dominate the first few principle components, this reduces the size of each colour ZIP code. Each input vector to the PCN consists of 3 individual ZIP code vectors, one for red, green and blue, each of 27 elements.

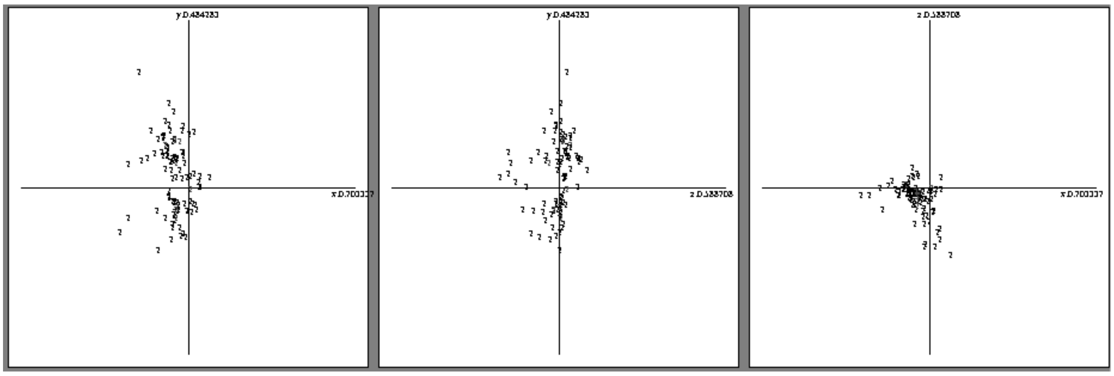


• Figure 5: Sample patches.

The ZIP codes from these 500 patches were then presented to the PCN and the 3D output space, plotted as three 2D spaces, was monitored over time as the network evolved. After a 500 iterations the output space had achieved the form shown in Figure 6 to Figure 9. In these images the 1st & 2nd, 2nd & 3rd and 1st & 3rd principle components are plotted respectively from left to right. Each point shown in this 3-space represents the output of the of the three PCN nodes for a given input ZIP code. The software allows the user to select a particular subset of points in order to observe the clusters formed by a specific image feature. It is also possible to highlight a point in the cluster and extract the image from which the associated patch was extracted. This allows a detailed assessment of the structure of the individual clusters.



• Figure 6: Wall to skirting board transition.



• Figure 7: Floor to skirting board transition

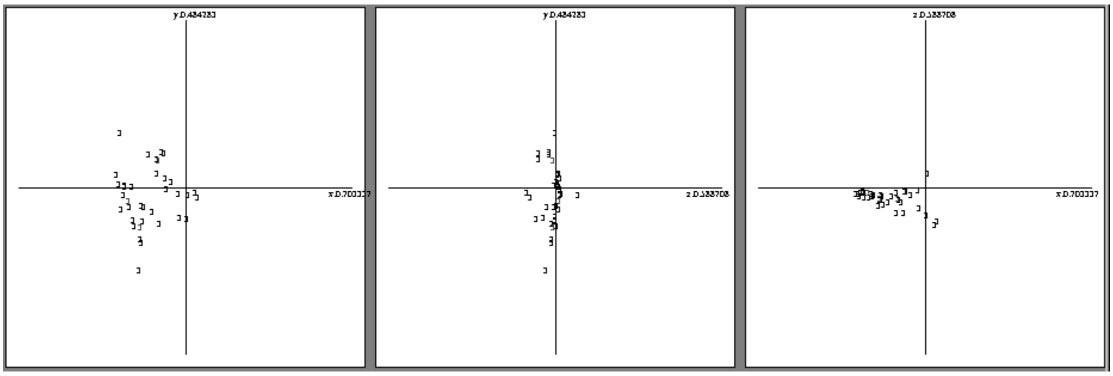
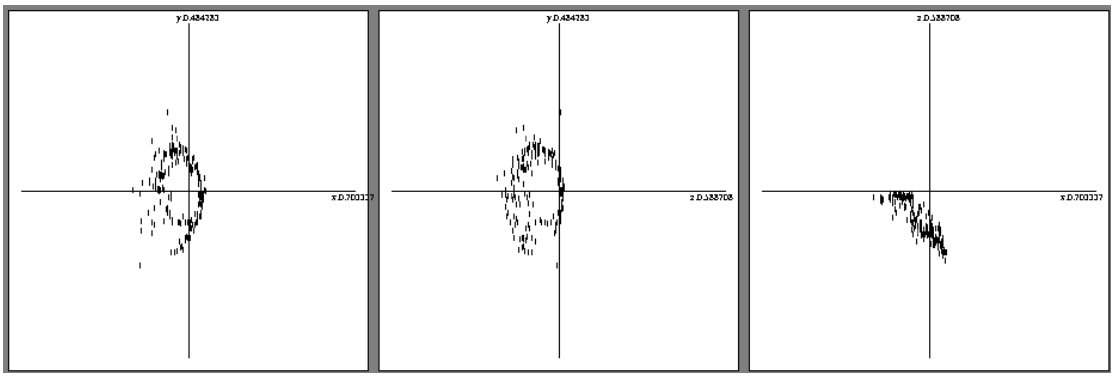
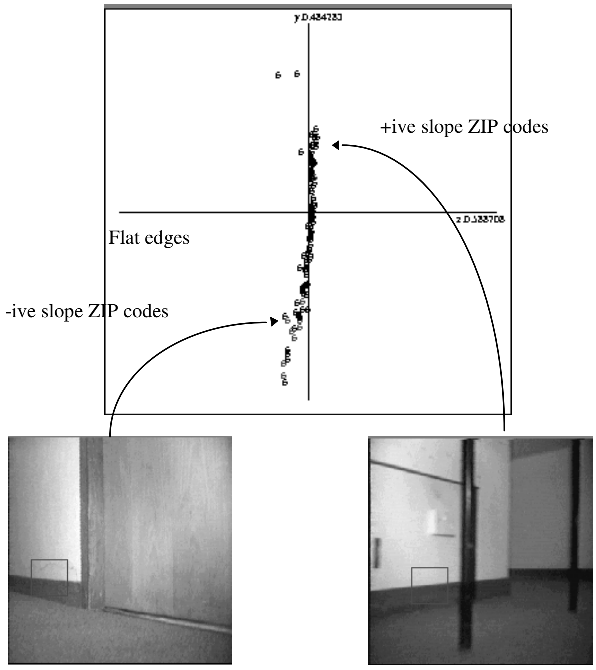


Figure 8: Floor to Wooden door transition.



• Figure 9: Floor.

Figure 6 to Figure 9 show typical results after 500 exposures to the set of ZIP codes. The different ZIP code sets have clearly formed clusters in the output space of the network. These clusters have self-organised as the PCN was not given any information about what each patch contained. For example the floor to skirting board cluster is very distinct from the wall to skirting board cluster. When these clusters are examined in more detail more interesting features can be observed. For example the skirting board cluster Figure 6 is organised in terms of the angle of the board in the image, such that left hand skirting boards, which tend to slope right are all stored at the bottom of the cluster, and right hand skirting boards, which tend to slope left are stored at the top, the tiny cluster in the middle of the band represents a set of images where the pixel patch was extracted from a horizontal board. Figure 7 also shows this with two distinct sub-clusters. Figure 10 identifies individual samples with their locations on the cluster.



• Figure 10: Directional sensitivity in cluster.

A similar effect can be observed in the floor cluster where one part of the cluster corresponds to floor patches that contain elements from other parts of the image, and so tend to represent floor close to the edges of a corridor or door, while another part of the cluster is associated with clear floor patches.

It is interesting to note the similarity between this process of extracting orientation specific information and the formation of orientation specific cells in Linsker's networks.

These tests clearly show the power of the PCN to rapidly extract and self-organise features from the ZIP codes however their interpretation relies on knowing which patch corresponds to which feature, and this in turn relies on the patches having been extracted by hand from pre-set locations in the input images, these are techniques that are not available to a system designed to operate in an unknown environment.

Using principal components for navigation

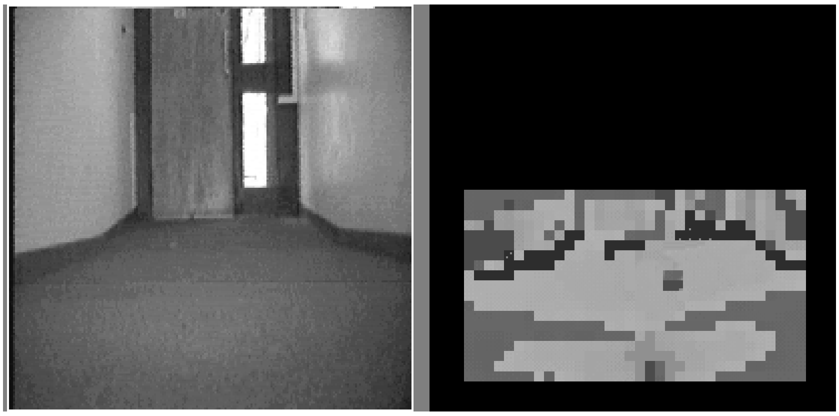
The previous results demonstrate the potential that a PCN has for self-organising feature extraction from pre-processed images. The following problems must still be solved if this is to prove to be a valid approach to providing visual navigation:

- There must be a way of extracting and integrating information from the whole visual field.
- There must be some mechanism for associating outputs to motor actions.

In the previous experiments the patches were pre-selected to be at interesting points. In a real world navigation task no assumptions can be made about the nature of the scenes in the visual field which in turn makes it impossible to select places to take patches from. Instead a selection of points are taken from across the visual field, since the visual patches used in the first stage of the vision system are quite large it these sample points can be quite sparse. There are two possible ways of processing the ZIP codes from these points.

In the first architecture the ZIP code from each point is presented to a single PCN, output from the PCN is then noted for each of these points and motion commands linked to regions in the output space. In order to see how this PCN architecture responds to the image it is possible to interpret the three outputs of the PCN as the three colours red, green and blue. In this way the clusters will appear as colour regions in the output space of the network. Figure 11 shows this for a typical scene. For the purposes of black and white presentation in this paper the significant colours have been converted to distinct grey values. Each grey values corresponds to distinct cluster in the output space of the PCN. The skirting boards are clearly picked out in a darker grey shade (orange in the colour image) while the floor is in lighter grey (blue in the colour image), including the rise in the floor level towards the door.

This architecture is suitable for simple low level navigation tasks where the segmentation of the scene



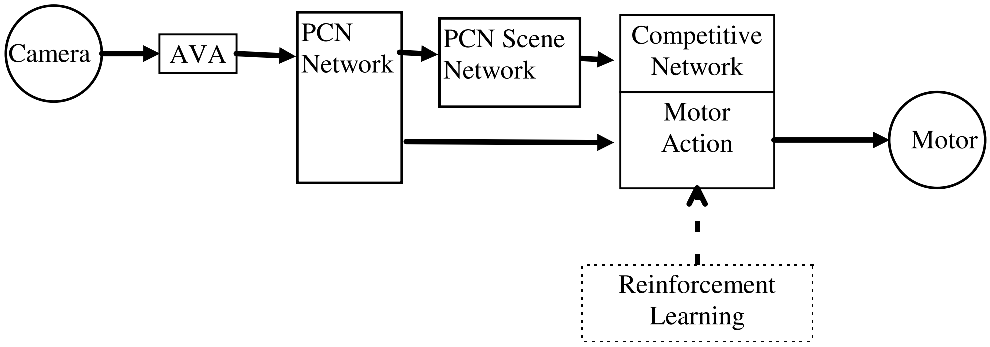
• Figure 11: region segmentation of image.

into relevant features is sufficient to guide the vehicle. So for example the skirting boards can act as repulsive forces in a reactive control scheme. It is important to note that this segmentation has been performed by a self-organising system with no information presented about the nature of the scene.

The second proposed architecture uses individual PCNs at each sparse sample point in the image. The outputs from these PCNs are then fed to a final stage PCN. This final stage PCN output represents a "scene space" in which clusters could form for "corridor ahead", "door in front" etc. In this case the individual point PCN adapt to their specific location, so for example the PCN on the outer left hand side of the visual field will see more left hand skirting board patches than the outer right hand one. This double level PCN allows motion decisions to be made based on local and scene level features.

Applying the VPCA to robot navigation Architecture (VPCA)

Before the system can be used in a complete navigation system a means of interpreting the output of the PCN in terms of motion commands is needed. This yields a navigation system with a general structure shown in Figure 12. From the above colour coding result clearly what is needed is an association between particular clusters in the output space and motor actions. Current work is trying to determine the best way of providing this. Possible solutions include the use of reinforcement learning mechanisms to associate motor actions to PCN output states. However it will be important to maintain the continuously evolving nature of the system rather than adopt a more discrete approach. Further work is also investigating how the network can be made to learn over a long time span without loosing features learned early in its exposure to data.



• Figure 12: Low level navigation block structure.

This structure would allow a robot to learn about new terrain but make best use of existing stored knowledge of previously encountered terrain. The non-object based nature of the system means that it should be able to adapt to any environment and any robot configuration with the addition of reinforcement learning based on the direct detection of danger signals, such as collisions, from additional sensors. This minimal system would be able to keep a robot operating and moving through unknown and previously unseen environments. Higher level navigation could be provided through the use of larger pixel patches matched on a location by location basis, and incorporated into a similar system to that identified above.

References.

[Mataric] "A Distributed model for mobile robot environment-learning and navigation" M.J.Matric MIT AITR 1228 1990.

[Nehmzow] *"Location recognition in a mobile robot using self-organisning feature maps"* U.Nehmzow, T.Smithers & J.Hallam. in Information Processing in Autonomous Mobile Robots, Springer Verlag 1991 pp 267-277.

[Brooks] *"A robust layered control system for a mobile robot",* R.A.Brooks IEEE Journal of Robotics and Automation 1986 pp14-23.

[Rao] *"An Active Vision Architecture based on iconic representations"* R.P.N.Rao D.H.Ballard. University of Rochester Computer Science TR548 1995.

[Smithers] *"On why better robots make it harder"*, T.Smithers, From Animals to Animats 3, 1994 pp64-72

[Walton] *"The Gello Machine: A self-organising neural network with modifiable architecture"* L.C.Walton Ph.D. Thesis University of Kent 1994.

[Horswill] *"Specialisation of perceptual processes"* I.Horswill MIT ATTR1511. 1994

[Kohonen] *"Self-organisation and Associative Memory"* T. Kohonen Springer Verlag 1988.

[von der Malsburg] *"Self-organisation of orientation sensitive cells in the striata cortex"* von der Malsburg Kybernetik, 14:85-100, 1973

[Linsker] *"From basic network principles to neural architecture: Emergence of orientation columns"* R. Linsker, Proc. Natl. Acad. Sci. USA, 83:8779-8783, 1986

[Plumbley] *"On information theory and unsupervised neural networks"* M. D. Plumbley Technical Report, Cambridge University Engineering Department, Cambridge University, 1991.