# Multiresolution Image Analysis Based On Local Symmetries

Andrew King, Roland Wilson
Department of Computer Science,
University of Warwick,
Coventry

September 28, 1993

**Abstract**

This report describes two seemingly distinct areas of work. First, there is the derivation of a new multiresolution image restoration algorithm, and secondly there is the development of a new approach to feature extraction. A review of these two areas is presented, the new algorithms outlined, and some results presented. The unifying principle behind this work is the idea of *image symmetries.* The background of this concept is reviewed, and its applicability to the current work explained. Finally, some speculations and expectations concerning the future direction of this research is included.

# Contents

# List of Figures

# 1 Introduction

The research described in this report can be divided into two separate areas. First, there is the work done on the restoration of noisy images, and secondly the analysis of shape features.

This work is unified by the concept of *symmetry.* Symmetry can be defined as the invariance of an image under a particular transformation. A transformation can therefore be referred to as a symmetry with respect to an image or a region of an image, if the result of its application upon the image block is to leave it unchanged. We deal with the idea of symmetry in two main related areas. First there is the role of symmetry in perception, and secondly symmetry as a property of images, and as a method of representation.

A common approach utilized throughout is that of multiresolution image processing. A brief description of this method is also included.

## 1.1 Symmetry In Perception

Symmetry in perception centres around the notion of *invariant pattern recognition.* A pattern recognition process is said to be 'invariant' with respect to a particular type of transformation, if the process is *transparent* to any change in the pattern resulting from such a transformation. For example, a pattern recognition process would be invariant with respect to scaling if it could recognize similar patterns at differing scales. This can be seen to be a similar property to that of symmetry, in that rather than the image remaining invariant under a transformation, the recognition process itself will remain invariant. Gibson [13] was one of the first to consider the role of invariance in human vision.

### 1.1.1 Invariant Pattern Recognition

Owing to the nature of real world perception, it is evident that to be of any practical use, a pattern recognition system will need to be invariant with respect to a particular set of commonly occurring transformations. These are known as the two-dimensional affine transform group, and consist of rotation, translation and scaling. Several approaches to constructing a model of invariant pattern recognition with respect to these transformations have since been taken. For example, Lamdan *et al.* [26] have devised an algorithm for affine invariant object recognition based on the extraction of 'interest points' from the image. The main drawback of their method though, is that it requires a model of each object that may be in the scene, a reasonable assumption in the industrial environment for which the algorithm was designed, but not so for many other real world problems. Simard *et al.* [31] have proposed a method whereby affine invariances are handled by using *a priori* knowledge about the derivatives of the function to be learned. This information is useful when we consider that for affine invariant recognition, the directional derivatives in the directions of particular transformations will be zero. Again though, the assumption that this information is known will not always be valid in real world perception problems. Teague [32] outlines a method of image analysis based on *image moments*. He shows that an image can be reconstructed from its set of moments, and that affine invariance can be achieved by 'normalizing' these moments.

As far as representing a model of mammalian vision is concerned though, little is claimed for these methods. We now examine one important attempt at such a model.

### 1.1.2 The Lie Transformation Group Model

The concept of image symmetry is central to the Lie Transformation Group (LTG) model of visual perception. This model was first formulated by Hoffman [16] as long ago as 1966. Its claim is that it can explain how the locally smooth phenomena that occur in the visual cortex can lead to a model of human vision.

As Dodwell [11] argues, the information that our eyes receive is in a continual state of flux owing to the motion of our head and body. The majority of this movement in our sensory input will be provided by transformations from the two-dimensional affine transform group. It is evident then, that in order for there to be any continuity in perception of objects over time, the role of such transformations in human vision must be central.

#### The Three Basic Lie Operator Pairs

To account for the role of transformations mentioned above, the LTG model proposes a set of three basic pairs of operators, or transformations, which it claims can be used to construct a model of human vision. These three pairs come in the form of mutually orthogonal *orbits* (smooth curves), and can be expressed in simple mathematical terms. These orbits are shown in Figure 1. The operators are said to be *natural*, in that the pattern recognition process should be transparent to motion in the direction of these orbits. Note that it is not claimed that the perceiver will not be aware of motion in these directions, just that such movement will not significantly impair the operation of the recognition process.

The first of the operator pairs is the simple translation, in the $x$ or $y$ direction. This is illustrated in Figure 1(a). This pair is part of the two-dimensional affine transform group, and so would seem to be a fundamental requirement of any visual model, for the reasons stated above, namely that in a world in which both the perceiver and the perceived are in constant motion, there is a need for continuity in object recognition over time. It should be noted that the translation operator will not always be used in isolation, but may be used in conjunction with one or more other operators (see

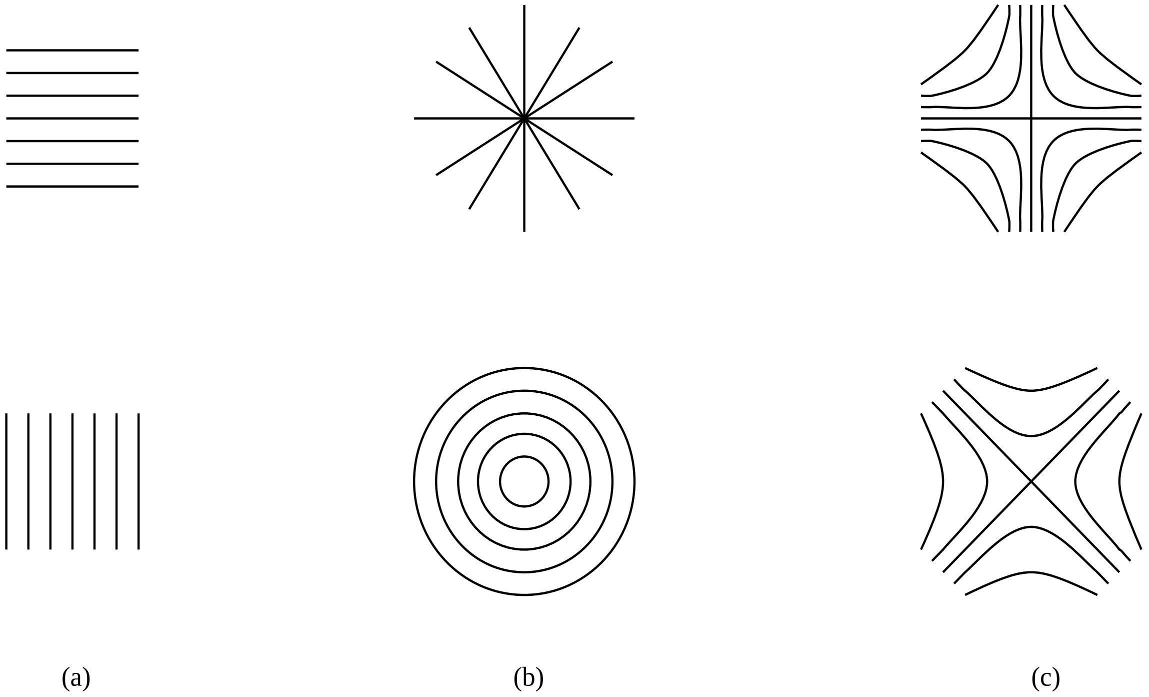

Figure 1: Orbits of the Three Pairs of Primitive Lie Operators

below). The $x$ and $y$ translation operators can be defined as

$$\mathcal{L}_X = \partial/\partial x \tag{1}$$

and

$$\mathcal{L}_\gamma = \partial/\partial y \tag{2}$$

As well as lateral movement, perceived objects in the real world may also move longitudinally, or towards or away from the perceiver. In addition, they may rotate relative to the perceiver. The orbits for the pair of Lie Operators for these transformations (the dilation and rotation) are shown in Figure 1(b). These are again part of the two-dimensional affine transform group, and can be defined as

$$\mathcal{L}_S = x(\partial/\partial x) + y(\partial/\partial y) \tag{3}$$

and

$$\mathcal{L}_O = -y(\partial/\partial x) + x(\partial/\partial y) \tag{4}$$

The orbits for the third, and for the moment, final pair of Lie operators, are shown in Figure 1(c). Their relationship to real world visual processing is not as immediately apparent as that of the first two pairs, and indeed their inclusion was suggested more by the mathematical nature of the other two primitive operators, than by intuitive reasons. The precise details of their derivation are not important, but Hoffman [16] interprets their inclusion as being related to the properties of binocular space. This final pair can be expressed as

$$\mathcal{L}_\beta = x(\partial/\partial x) - y(\partial/\partial y) \tag{5}$$

and

$$\mathcal{L}_b = y(\partial/\partial x) + x(\partial/\partial y) \tag{6}$$

It would be useful now to take a brief look at the way in which the Lie operators are derived. This is done by using the relationship between the equation $F$ that describes the trajectory of a Lie orbit, and its corresponding Lie operator $\mathcal{L}$. This relationship can be defined as

$$\mathcal{L}F = 0 \tag{7}$$

To take an example, consider $F$ to be the horizontal line $y - b = 0$. As its corresponding Lie operator is $\partial/\partial x$, it can be clearly seen that the result of $\mathcal{L}F$ in this case is zero. We will return to this later.

The basic postulate of the LTG model of human vision, is that the structure of the visual cortex is determined by these basic Lie operators. It should not, however, be supposed that these three pairs are the only ones that can be derived. Many more could be discovered, especially when temporal properties are considered. This would provide a new differential operator, $\partial/\partial t$, in addition to $\partial/\partial x$ and $\partial/\partial y$. They are however, the three which would be the most common and useful in a two dimensional visual processing problem. Since the structure of the visual cortex is to a large degree determined by the information it receives from its environment, it would seem natural that these three basic pairs will be predominant.

### Pattern Recognition

It should be noted here that there are certain image features that will remain invariant under transformations corresponding to certain of the Lie operators mentioned earlier. For example, a straight edge will not be affected by a particular combination of the translation operators shown in Figure 1(a). For this reason, we refer to these transformations as local symmetries with respect to the corresponding local image features.

Dodwell [11] goes on to outline a process by which pattern recognition could occur within the framework of the LTG model. The fine details of this are not explained, but it should be noted that it centres around the idea of *cancellation*. That is to say, the visual system will search for which operator(s) will minimize the output of any orbit. To put this another way, it will find the orbit whose Lie operator best satisfies Equation 7. This process could be said to be recognizing the characteristic local symmetries of the pattern in question. It should be noted however, that this process is limited by its strictly local nature and by the small number of shapes that can described simply in these terms. For this reason it would seem that as a global recognition process it is less than satisfactory, but as a general low level model of the underlying processes in human vision, it may be of some use.

### Neuropsychological Evidence

As Dodwell [11] has pointed out, a lot of structure has been discovered in the mammalian visual cortex [19], which may offer evidence in support of the LTG model. In particular, it has been found [18] that receptive fields have certain vector-like properties which would be in keeping with the cancellation process described above.

In addition, Bigün [4] has noted the relevance of certain experiments ([21], [27]) in which circularly symmetric shapes (i.e. shapes invariant under rotation, see Figure 1(b)) were observed by subjects when low frequency magnetic fields were applied to their temples.

It does seem though, that to suppose that the LTG model is a complete model of mammalian vision would be optimistic. Hoffman [16] does not appear to place a high priority on the biological aspect of the model, and it is difficult to imagine that what he proposes is the complete story. It seems likely though, that Lie transformations may play a significant role in our visual processes.

## 1.2 Symmetry In Representation

Previously we have discussed the role of symmetry in image processing. A similar property to symmetry is that of *self-similarity*. An image can be said to possess self-similarity when the image as a whole resembles a scaled up version of a part of itself. It has been pointed out [28] that this could be seen as a special case of symmetry involving scaling operations. We will return to this later.

For now though, it was noted earlier that certain image features will have different characteristic local symmetrical transformations. The local symmetries of images could be said to 'represent' the image in question. It would seem then, that knowledge of the characteristic local symmetries of an image would be of use in trying to construct a method of image coding. We now examine one such method.

### 1.2.1 Block Based Fractal Coding

The technique of Block Based Fractal Coding of images was first developed by Jacquin [20]. The basic concept behind the technique is that for every image there exists a transformation (the fractal transformation) upon that image, for which the result of its application upon the image will be to leave it unchanged. It can be seen immediately that there are similarities here to the idea of local image symmetries discussed earlier. The image in question can be encoded as its fractal transformation.

Furthermore, the result of applying this transformation to any other image will be to perform a *contraction mapping*, or to bring it closer by some measure of similarity to the image to be encoded.

### The Contraction Mapping

To expand upon this, we must first make clear the idea of a contraction mapping. As stated above, a contraction mapping is a function which increases some measure of similarity to an *ideal* image. To put this another way, it is a function which reduces a *distance measure* between the domain image and the ideal image. If we define a distance measure between two images $x$ and $y$ as $d(x, y)$ then for the affine transformation $T(x) = Ax + b$ to be a contraction mapping, the following equation must hold.

$$d(T(x), T(y)) \leq L.d(x, y). \tag{8}$$

where $L$ is a real number between 0 and 1. The specific distance measure used can vary, and the details of such measures are not currently relevant.

We illustrate this with an example from Barnsley [2]. He provides several sets of affine transformations, each with an associated probability, which can be used to construct particular images, using what he calls the Random Iteration Algorithm. Each of these sets of transformations constitute a contraction mapping with respect to the image they represent. The algorithm works as follows. Starting with an image of a single point, the transformations are iteratively applied, with the particular transformation used in each iteration being dependent on the probabilities. This will build up successive images of gradually increasing numbers of points. These successive images are of a correspondingly greater complexity, although for a digital image a limit will be reached at which the image will no longer change. Therefore these transformations together constitute the 'fractal transformation' of the image in question.

An example of this is shown in Figure 2. This illustrates the Random Iteration Algorithm run for varying numbers of iterations using the fern leaf set of transformations. The top left image is the result after 100 iterations, the top right is after 500 iterations, the bottom left after 1000, and the bottom right after 10000 iterations. As can be seen, the complexity of the image grows in proportion to the number of iterations applied. This image can also be said to possess a large amount of self-similarity, as defined earlier. To a certain extent, all images possess a degree of self-similarity, and the task of the encoding process (see below) is to identify ways in which this can be expressed.

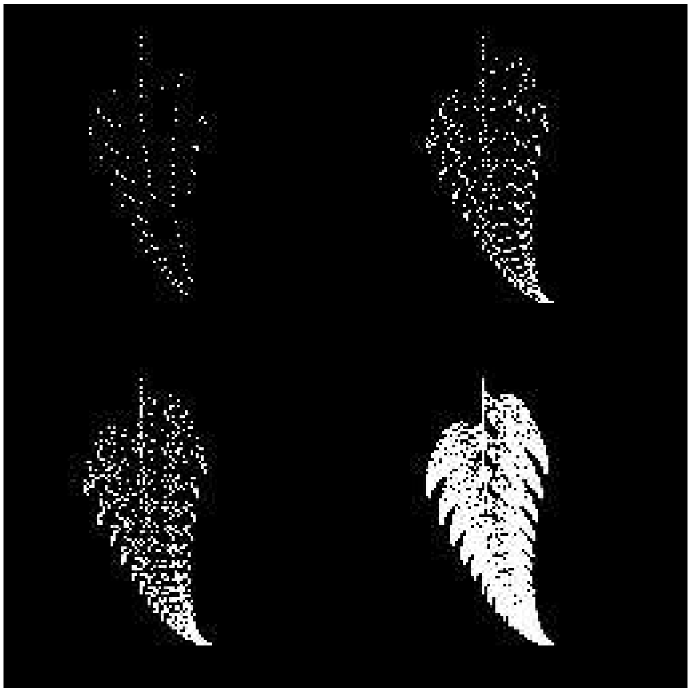

Figure 2: Results of the Random Iteration Algorithm

### The Encoding Process

We will take a brief look at the way in which images can be encoded using this technique. The fractal transformation for an image is determined from a series of local transformations, each operating on a *block* of the image [20]. The domain image is divided up into a set of non-overlapping image blocks, and the range image is correspondingly divided into a set of blocks, each of which will be half the size of those in the domain image. The elementary transformations will then operate on these blocks. The precise nature of these transformations is not immediatedly relevant, but the important point to note is that they will consist of a combination of two-dimensional affine transforms. Comparisons can be drawn here between the calculation of the fractal transformation of an image, and the Lie Group pattern recognition process described earlier. Both consist of combinations of two-dimensional affine transforms, but the role of these transforms differs in each case. In the Lie group process, the transformations were used in a cancellation operation in order to recognize the local image structure, whereas here we are trying to describe this structure by finding the appropriate symmetrical affine transformations.

From the above, it follows that any image for which a fractal transformation can be determined, can be encoded as this transformation, and that if the transformation can be stored so as to take up less space than the original image, then the technique could be a useful tool in data compression. Impressive results have been claimed for this technique [20].

## 1.3 The Multiresolution Approach to Image Processing

It was mentioned in Section 1.1.1 that it is important that a pattern recognition process should be invariant to certain transformations. These transformations include translation and scaling. In order for this invariance to be achieved we must find a way of representing the image in such a way as to be able to examine it at differing positions and scales. Such a representation is called a *multiresolution* representation.

### 1.3.1 Review

One such approach that can be of use in facilitating scale invariance is the *wavelet transform* [10]. This is a way of analysing the frequency content of a signal locally in time. In this sense it is quite similar to the Windowed Fourier Transform. The wavelet transform can be used to considerably simplify the recognition of scaled patterns. To expand upon this, it can be shown that the wavelet transform of a scaled pattern is merely the wavelet transform of the unscaled pattern with a simple co-ordinate translation within the transform domain. To express this more formally, we look at a one-dimensional signal $f(t)$, and define its wavelet transform to be $T^{wav}_{m,n}(f)$. The parameter $m$ indicates the scale required, and $n$ indicates the time locality required. If $f$ is then transformed by a scaling operation $T = Ax + B$ $(A > 0)$, the wavelet transform of the transformed signal can be defined as $T^{wav}_{Am,An+B}(f)$. This effectively reduces the scaling operation to a simple translation (in the transform domain), thus making invariant pattern recognition significantly easier. This principle can be extended to two-dimensional signals.

Another relevant multiresolution approach is that of scale-space filtering, developed by Witkin [35]. In this technique, a one-dimensional signal is first convolved with a second partial derivative of a Gaussian filter at various scales. The *scale-space* of the signal is then defined as being the set of all *zero-crossings* of this signal. This approach has also been discussed in depth by Koenderink [23]. As Bischof [5] has pointed out though, there are problems when it comes to generalizing this to two-dimensional images. In addition, the fact that these methods feature a continuous variation of the scale parameter pose a practical 'implementation' problem.

The above examples help to illustrate another potential use of multiresolution approaches, notably that they can also be of use in identifying image features, as most important features occur fairly consistently across scale. In addition, it is a common property of natural and artificial images that there are features present in the image which will vary with respect to their scale. Multiresolution approaches also allow a greater flexibility in the analysis process, in that different regions of the image can be analysed at different resolutions, and in that no decision about which scale should be used for each region need be made in advance.

A significant example of the multiresolution approach which is utilised in the work described in this report is the *image pyramid*. This is now dealt with below.

### 1.3.2 The Image Pyramid

The image pyramid has many similarities with the wavelet transform described earlier, and can also be viewed as a discrete form of scale-space representations. It should be noted that this use of discrete rather than continuous forms poses problems in terms of the decimation and interpolation operations which implement the change of scale, in that a certain amount of aliasing can be introduced, leading to a lack of circular symmetry [34].

The multiresolution approach takes into account the requirement for invariance over scale by representing the image at several different spatial resolutions. This necessarily introduces an element of redundancy, but this is a relatively small price to pay for the advantages it affords. In this technique, the original image is sub-sampled to create a set of images known as a *pyramid*, starting with the original image, and with each successive image at a lower resolution (usually half) than its predecessor. Originally, most pyramids were built in a quad-tree structure, with each sub-sampled pixel having four children, and each child only one parent. In this case the parents grey level would be set to the average value of its four children. However, Burt and Adelson [8] proposed the use of a Gaussian weighting function to replace the quad-tree approach, in which for a 5 by 5 filter each parent has 25 children, and each child 9 parents. To state this more precisely, if $w(m,n)$ is a Gaussian weighting function, and $g_l(i,j)$ the image at level $l$ of the pyramid, then the construction of the pyramid can be defined as

$$g_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) g_{l-1}(2i+m, 2j+n). \tag{9}$$

This method has the effect of lowpass filtering each successive image as the pyramid is built, and has the advantage of reducing artefacts created by aliasing. It does however, introduce an overhead in the amount of computation required, with 25 multiplications required for each pixel in the constructed level.

Once the pyramid has been built up to a certain level, a new sequence of images can then be reconstructed, back down to the bottom level again. This is done by interpolating new pixel values in the child image given the values in the parent image. This process can be defined by

$$g_l(i,j) = 4 \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n).g_{l+1}(\frac{i-m}{2}, \frac{j-n}{2}). \tag{10}$$

where only terms for which $(i-m)/2$ and $(j-n)/2$ are integers are considered. The result of constructing a pyramid in this manner is that we then have two sets of images, one generated 'on the way up' the pyramid, and one reconstructed 'on the way down'. It is apparent that the interpolated images will have been subjected to a lowpass filter and so will be smoothed.

The image pyramid has been used in many different problems in image processing. Some of the more important examples are reviewed below.

An early example of predictive image coding using a quad-tree pyramid structure was developed by Wilson [33]. In this work, the transmission of the coded image would commence with the image at the top level of the pyramid, which will consist of a single pixel. A succession of images would then be transmitted each being equal to the difference between a reconstruction of the data from the level above, and the actual image at the current level. By using this method, the receiver of the image can construct a prediction of the transmitted image, which will then be refined by each successive transmitted image.

Another example which was applied to the problem of image restoration is that by Clippingdale [9]. In this work, a quad-tree pyramid was built starting with a noisy image at its base, and ascending the pyramid up to the top, where there is a single node containing the average grey level. This structure was then used to construct an estimate of the original clean image, by combining information vertically in the pyramid.

Bhalerao [3] provided a further example, this time concerned with the problem of image segmentation. This approach used combined a boundary estimation process with a region estimation process, the results of which are iteratively improved.

### 1.3.3 Neuropsychological Parallels

Neuropsychological parallels to the multiresolution approach centre around the notion of scale. As Hubel [17] has pointed out, there are cells present in the mammalian retina and visual cortex that respond to features at several scales. This would seem to tie in well with the analysis over several scales that occurs in multiresolution image processing. Indeed, it is difficult to imagine a complete visual model which did not incorporate some notion of scale. This would seem then, to offer some encouragement for the use of multiresolution approaches to image processing, although it should be noted that no claims are being made as to the biological plausibility of the model.

# 2 A Multiresolution Approach To Image Restoration

## 2.1 Review

Much work has been done in the past in the field of restoration of images degraded by noise. The task of removing noise from a purely uniform image would be fairly straightforward, a simple smoothing (averaging) algorithm would eliminate the degradations. However, although images do often contain large areas of uniformity, they will also feature areas of higher grey level variation, or edges. In these areas a simple smoothing operation will result in a loss of information. It is clear then, that a different sort of processing will be required in these areas. The problem which has faced researchers therefore, can be split into two main sub-problems. Firstly, there is the problem of knowing where the lower frequency areas of the image are, in order that smoothing algorithms can be applied to remove the noise, and secondly there is the question of what to do in the higher frequency areas.

Smoothing of images whilst preserving the edges has many important applications, and so has been a popular area of research in previous years. A useful review of existing averaging techniques was given by Du Buf and Campbell [7]. The filters they considered used various techniques to overcome the problem of what to do at edges. A common method is to take the median rather then the mean of the local neighbourhood of pixels. This kind of approach is more likely to reduce the amount of blurring in the boundary regions than normal mean averaging. In addition to this, some methods attempt to make some sort of allowance for the orientation of the edge (e.g. [15]), although the techniques employed to do this are rather unsophisticated.

As was described in Section 1.3.2, Clippingdale [9] has adopted a multiresolution approach to noise reduction. Essentially, this work still uses a form of averaging to achieve the noise reduction, but the determination of the boundary areas differs in the inclusion of a notion of scale.

## 2.2 The Algorithm

The new restoration algorithm described in this section could be said to be conceptually similar (although not the same) to the median approaches referred to in Section 2.1, in that all that is happening at the edges is that one pixels grey level value is being replaced by anothers. However, the reasons for taking this direction were somewhat different.

To understand the motivation for the approach taken in this work, we first need to refer back again to the concept of local symmetries mentioned earlier. It was stated then that a transformation is a symmetry with respect to a part of an image if its application leaves that part of the image unchanged. It can further be stated that applying the transformation to any other image block will bring that block 'closer' to the image block for which the transformation is a symmetry. That is to say, the transformation is a contraction mapping as described in Section 1.2.1. Therefore, if we knew the appropriate local symmetries for an image, then the restoration process (whether from a noisy or blurred version) would be easy.

The basic data structure used is a Gaussian filtered image pyramid similar to the one described in Section 1.3.2. Use of this structure will automatically have the effect of smoothing the image, owing to the reduction in information further up in the pyramid. The amount of smoothing will depend upon how many levels there are in the pyramid.

Certain processing will be performed on the images as they are reconstructed from higher levels in the pyramid. The aim of this processing will be to minimize the Mean Squared Error (MSE) between the pairs of images at the same level of the pyramid (the *generated* and *reconstructed* images). Whenever MSE is referred to from here on, it will refer to this value.

The basis of this processing is the application of a set of translations in the edge regions with the aim of *sharpening up* the blurred edges. It should be clear that if these translations are applied *towards* the notional edge, such a *sharpening* will occur, thus reducing the MSE. This idea is conceptually similar to the idea of local symmetries discussed in Section 1, in that a clean version of the edge would remain invariant under these translations.

### 2.2.1 Orientation Estimation

To start with though, we must make some sort of estimation of the directions in which the translations should be applied. This is done by a fairly simple process.

First, the reduction in MSE produced by applying each of four primary translation directions is calculated. These are NE, NW, SE and SW. From these values, four vectors can be derived in these four directions. A simple addition of these vectors is enough to provide a rough estimate of the desired translation direction at this point.

This orientation is represented using a *double angle* format, a method first employed by Granlund [14]. The doubling of the angle effectively treats opposite angles (differing by 180 degrees) as being identical, removing the phase shift as the boundary is crossed. The advantage of this representation is that it allows us to *smooth* the orientation vectors without the opposing vectors in the edge regions cancelling each other out. This smoothing can help to counter the effects of noise in the orientation estimate.

### 2.2.2 Vertical Propagation

In order to take full advantage of the multiresolution model however, we should note that owing to the lower amount of noise higher up the pyramid, the orientation estimation will be more accurate the higher up we go. Bhalerao [3] has outlined a method by which information can be propagated downwards to improve the accuracy of the orientation estimates further down the pyramid. This can be defined as

$$\hat{\theta}_{i,j}(l) = \beta_{i,j}(l)\hat{\theta}_{i/2,j/2}(l-1) + (1 - \beta_{i,j}(l))\tilde{\theta}_{i,j}(l). \tag{11}$$

where $\hat{\theta}_{i,j}(l)$ is the double angle orientation estimate at level $l$ after vertical propagation, and $\tilde{\theta}_{i,j}(l)$ the double angle estimate before propagation. $\beta_{i,j}(l)$ is the *feedback* coefficient for level $l$, and is a measure of how much information is propagated downwards in the pyramid. The calculation of the feedback coefficient at each level is based upon estimation of the signal and noise variances on the respective levels, and is outlined in [3].

Once a reliable orientation estimate has been arrived at, a decision must then be made as to whether to perform a translation at a given pixel, or whether to leave it unchanged. Since the orientation is held in a double angle format, there are two possible directions for the translation to be performed in. The decision is based on a simple thresholding mechanism. First, the reduction in MSE resulting from each of the two possible translations is calculated, then if the greater of these exceeds a given threshold, that translation is performed. The actual threshold used was equal to the MSE for the level above the current level.

In order to counter the effects of noise, an improvement in performance can be obtained by looking at a local neighbourhood of pixels, and averaging the reduction in MSE over these pixels. This necessarily incurs an overhead in computational cost however. A 3 X 3 neighbourhood was found to produce a significant improvement without too great an overhead.

## 2.3 Computational Requirements

The major computational cost of this algorithm lies in the multiplications needed to compute the MSE at various points. The number of multiplications necessary per image pixel is summarized in Table 1. This data was derived on the assumption that a 3 X 3 window was being used to make the translation decision.

## 2.4 Results

The restoration algorithm was run on the shapes image, with added white Gaussian noise of several different variances. A summary of the results is shown in Table 2.

In order that the subjective improvement can also be gauged, the results for the 6dB image are shown in Figures 3 and 4.

These results compare favourably with those achieved by Clippingdale [9]. He provided two sets of results using the 'blobs' images (an artificial image comparable to the 'shapes' image used here). The first of these was for the use of an isotropic

| *Process* | *No. Multiplications* |
|---|---:|
| Orientation Estimation | 4 |
| Vertical Propagation | 2 |
| Translation Decision | 18 |
| Calculation of MSE | 1 |
| Estimation of Variances/Covariance | 4 |
| Total | 29 |

Table 1: The Computational Cost (Per Image Pixel) of the Restoration Algorithm

| *Input SNR* | *Output SNR* |
|---|---:|
| -12dB | 9.7dB |
| -6dB | 14.8dB |
| 0dB | 17.5dB |
| 6dB | 19.5dB |
| 12dB | 20.5dB |

Table 2: The Improvements Made on a Noisy 'Shapes' Image

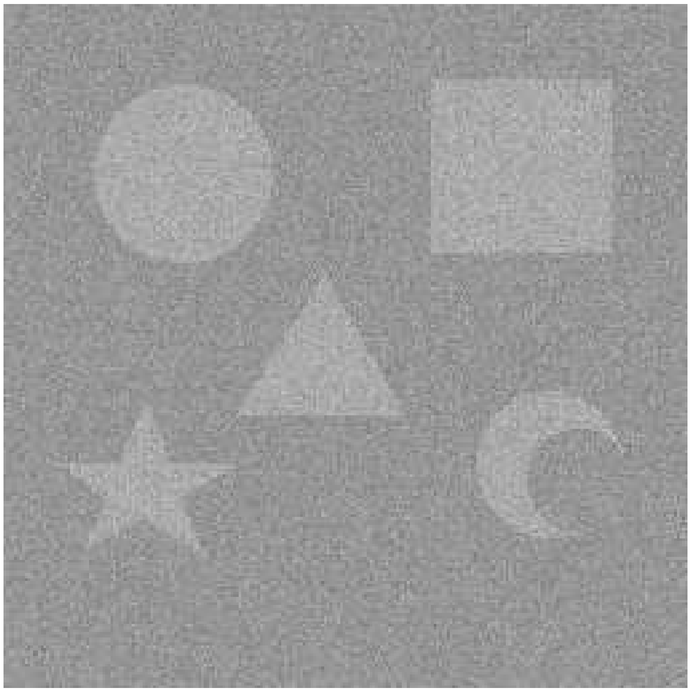

Figure 3: Noisy Shapes Image (SNR = 6dB)

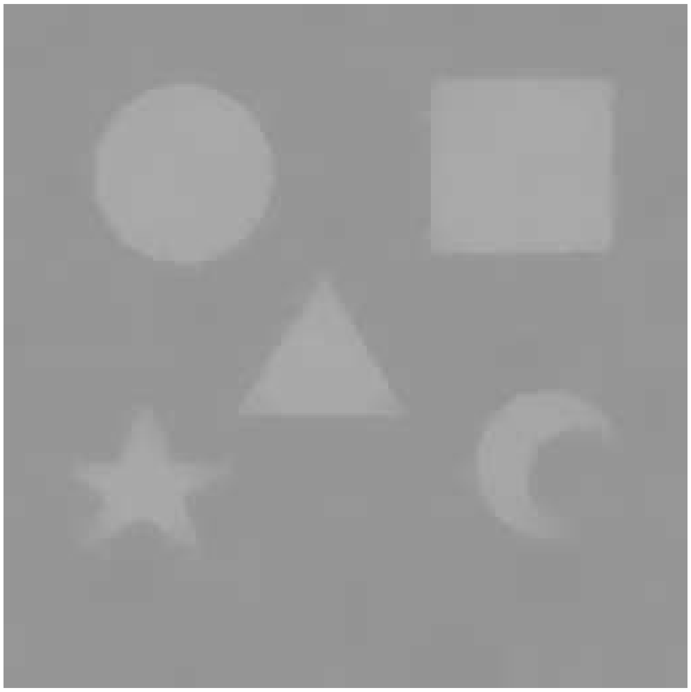

Figure 4: Restored Version of Noisy Shapes Image (SNR = 19.5dB)

filter, and the second for the use of an anisotropic filter. For a -12dB image, the restored versions were 9.5dB and 9.6dB respectively, for the 0dB image, they were 15dB and 14.6dB, and for the 12dB image, they were 22.1dB and 19.4dB.

Overall then, the results appear to be quite impressive, especially at straight edges. Performance at corners of shapes seems to be comparatively poor, with a certain amount of degradation owing to blurring. At edge regions the algorithm will *sharpen up* this blurring, but it seems to have difficulty in doing this at areas of high curvature (i.e.corners). We will return to this problem in Section 4.2.

# 3 Bilinear Operators for Local Symmetry Detection

## 3.1 Review

Feature extraction methods have traditionally concentrated on simple image features such as edges. It is obvious though, that any model of human vision will need to incorporate processes to analyse higher level features.

One example of such a higher level feature of both natural and artificial images is curvature. Previous work on the detection of curvature in images can be divided into two distinct categories. Firstly, there is the use of one-dimensional operators. These methods take a previously determined edge map, and process this in some way (e.g. differentiate it twice) to give a measure of the amount of curvature present (e.g. [30], [6]). A slightly different approach has been taken by, for example, Freeman [12] or Anderson [1], in which *steerable* filters are used to obtain information about possible multiple orientations in one region, indicating the presence of complex neighbourhood features such as corners. Also, a scale based method was proposed by Rattarangsi and Chin [29]. These methods are still similar to the one-dimensional approaches in that we must have *a priori* knowledge of edge information. It has been pointed out [24], that if we are concerned with the problem of human (or mammalian) vision, then the inherent sequentiality of such methods makes them less biologically plausible (see Section 3.3).

The second category, that of the two-dimensional operators, operate directly on the grey level image. This approach has not been as common as the one-dimensional method, but an example of it can be provided by Koenderink (e.g. [24], [25]). He outlines a process which combines various derivatives of the luminance image to yield a measure of curvature. Interesting though this work is however, it is difficult to evaluate its effectiveness without any experimental results provided.

A more promising and relevant approach is taken by Bigün [4]. His work incorporates the idea of local symmetries referred to earlier. He proposes a technique which can be used to obtain a measure of certainty as to whether a local image neighbourhood is *circularly* symmetric. By *circularly* symmetric he means symmetric with respect to rotation and/or scaling (i.e. an area of curvature). Again though, the method consists of a complex filter operating on a partial derivative image, thus introducing an element of sequentiality into the process. He does however go on to deal with *central* (i.e. dilation) and *linear* (i.e. translation) symmetries, and applies these to the problem of pattern recognition. The neuropsychological plausibility of this approach was discussed in Section 1.1.2.

## 3.2 The Feature Extraction Operators

As was noted earlier, it is possible to describe the structure of an image in terms of its various spatially distributed local symmetries. Bearing this in mind, it would be of use if we had a way of determining these local symmetries. A simple and elegant way of doing this would seem to be the use of a set of two-dimensional filters to detect shape features.

The technique used to detect shape features consists of applying a pair of 3 X 3 filters developed by Wilson and Bhalerao [34]. Each of these filters determine the first partial derivative of the image, and are oriented in mutually orthogonal directions. These orientations are at 45 degrees to the $x$ and $y$ axes, we shall call these axes the $u$ and $v$ axes. This is illustrated in Figure 5. These derivatives can then be combined to determine two more first partial derivatives, oriented along the $x$ and $y$ axes, using the following relationships.

$$\frac{\partial f}{\partial x} = \frac{1}{\sqrt{2}}\left(\frac{\partial f}{\partial u} + \frac{\partial f}{\partial v}\right) \tag{12}$$

$$\frac{\partial f}{\partial y} = \frac{1}{\sqrt{2}}\left(\frac{\partial f}{\partial u} - \frac{\partial f}{\partial v}\right) \tag{13}$$

By applying the filters again we can determine second order derivatives for the $u$ and $v$ axes. The second order derivatives for the $x$ and $y$ axes are then determined using the following relationships.

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{2}\left(\frac{\partial^2 f}{\partial u^2} + \frac{\partial^2 f}{\partial v^2}\right) + \frac{\partial^2}{\partial uv} \tag{14}$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{1}{2}\left(\frac{\partial^2 f}{\partial u^2} + \frac{\partial^2 f}{\partial v^2}\right) - \frac{\partial^2}{\partial uv} \tag{15}$$

Various combinations of these first and second order partial derivatives constitute our feature extraction operators. The purpose of these operators is to signal the breaking of image symmetries in local neighbourhoods of the image. A typical pixel not in an edge or textured region can be characterized by noting that its neighbourhood can be replaced by one translated in an arbitrary direction, dilated about the pixel, or even rotated, without being changed significantly. Near edges, in particular, this is no longer true. The symmetry is broken. The aim of these operators is to identify in what way the symmetry has been broken. That is to say, is some subgroup of the symmetries still valid?

Four different combinations of these filters are performed. These are summarized below. In the following sections, the four orientations will be denoted by the axes $x$, $y$, $u$ and $v$. The image to be filtered will be called $f(x, y)$. Hence the first derivative of the image in the $x$ direction can be expressed as $\frac{\partial f}{\partial x}$, and so on for the other axes

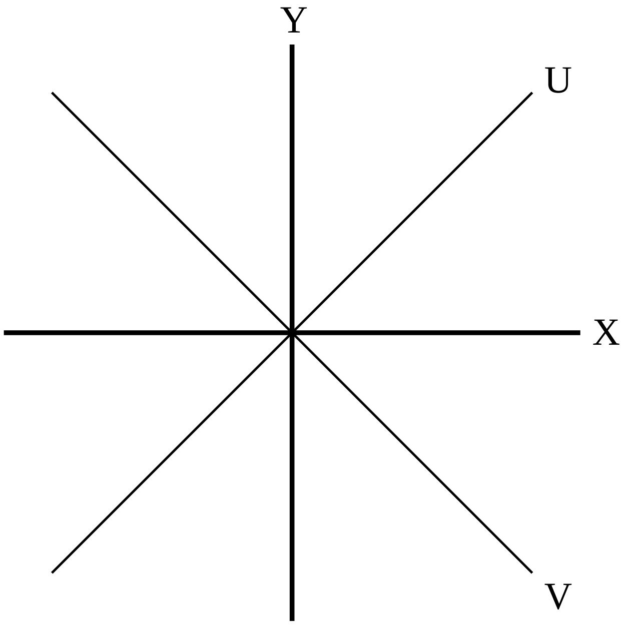

Figure 5: Four Orientations of the Partial Derivatives

and the second derivatives. The orientations of the four axes are $\theta_x$, $\theta_u$, $\theta_y$ and $\theta_v$, and their values can be seen from Figure 5 to be 0, $\pi/4$, $\pi/2$ and $-\pi/4$ respectively.

### 3.2.1 Translation

The most basic symmetry is translation, which can be represented as a two dimensional vector. Now if the image is not (locally) symmetric with respect to arbitrary translations, it may nonetheless be symmetric with respect to translations in a particular direction. Recalling the discussion of Section 1.1.2, we can see that this is equivalent to saying that there is some direction $\phi$ for which the gradient in that direction is zero. For smooth images, this amounts to selecting the axis which is perpendicular to the gradient direction. If $\vec{g}(x, y)$ is the gradient at $(x, y)$, then we need to find an orientation $\phi(x, y)$ such that

$$\vec{g}(x,y).\begin{pmatrix} \cos\phi(x,y) \\ \sin\phi(x,y) \end{pmatrix} = 0 \tag{16}$$

Now clearly if $\phi(x, y)$ is such an orientation, then so is $\phi(x, y) + \pi$. The solution to this ambiguity is the use of a double angle representation [14] (see Section 2.2.1). The double angle vector for the orientation is defined by

$$\vec{t}(x,y) = \frac{\partial f}{\partial x}.\frac{\partial f}{\partial y}\begin{pmatrix} \cos 2\theta_u \\ \sin 2\theta_u \end{pmatrix} + \frac{\partial f}{\partial u}.\frac{\partial f}{\partial v}\begin{pmatrix} \cos 2\theta_x \\ \sin 2\theta_x \end{pmatrix} \tag{17}$$

This combination of the first and second partial derivative filters is quite similar to one proposed by Knutsson [22].

In order to analyse the behaviour of this operator over edges at differing orientations, we define our image to be an 'ideal' edge of orientation $\varphi$ as illustrated in Figure 6.

$$f(x,y) = Erf(g(x,y)) \tag{18}$$

where

$$g(x,y) = x\cos\varphi + y\sin\varphi \tag{19}$$

The 'Erf' represents an *Error Function*, and is the result of convolving the supplied function with a circular Gaussian $e^{\frac{-(x^2+y^2)}{2}}$. It can be defined as

$$Erf(z) = \frac{2}{\pi}\int_0^z e^{-t^2}\,dt. \tag{20}$$

Now we can calculate the partial derivatives of Equation 18 (see Appendix A) and substitute them into Equation 17. This should gives us an equation in terms of $x$, $y$ and $\varphi$ which will express the response of the translation operator to the ideal edge image over various orientations. If this equation is evaluated and simplified, it can be expressed as

$$\vec{t}(x,y) = \frac{2}{\pi}e^{-2(x\cos\varphi+y\sin\varphi)^2}\begin{pmatrix}\cos 2\varphi\\ \sin 2\varphi\end{pmatrix} \tag{21}$$

This shows that the double angle returned should give an accurate indication of the orientation of the symmetry breaking feature.

### 3.2.2 Dilation

The dilation operator signals the local image neighbourhoods where other symmetries may have broken down, but the dilation symmetry still holds. Intuitively, we can see that this operator should respond to edges and corners. Its response is similar to the translation operator mentioned above (Section 3.2.1), with the exception that the angle of the vector returned is a single angle vector. For this reason, the phase shift should remain intact. The vector should point *towards* the edge, or *into* the corner. It can be expressed as

$$\vec{d}(x,y) = \begin{pmatrix}d_x\\ d_y\end{pmatrix} \tag{22}$$

where

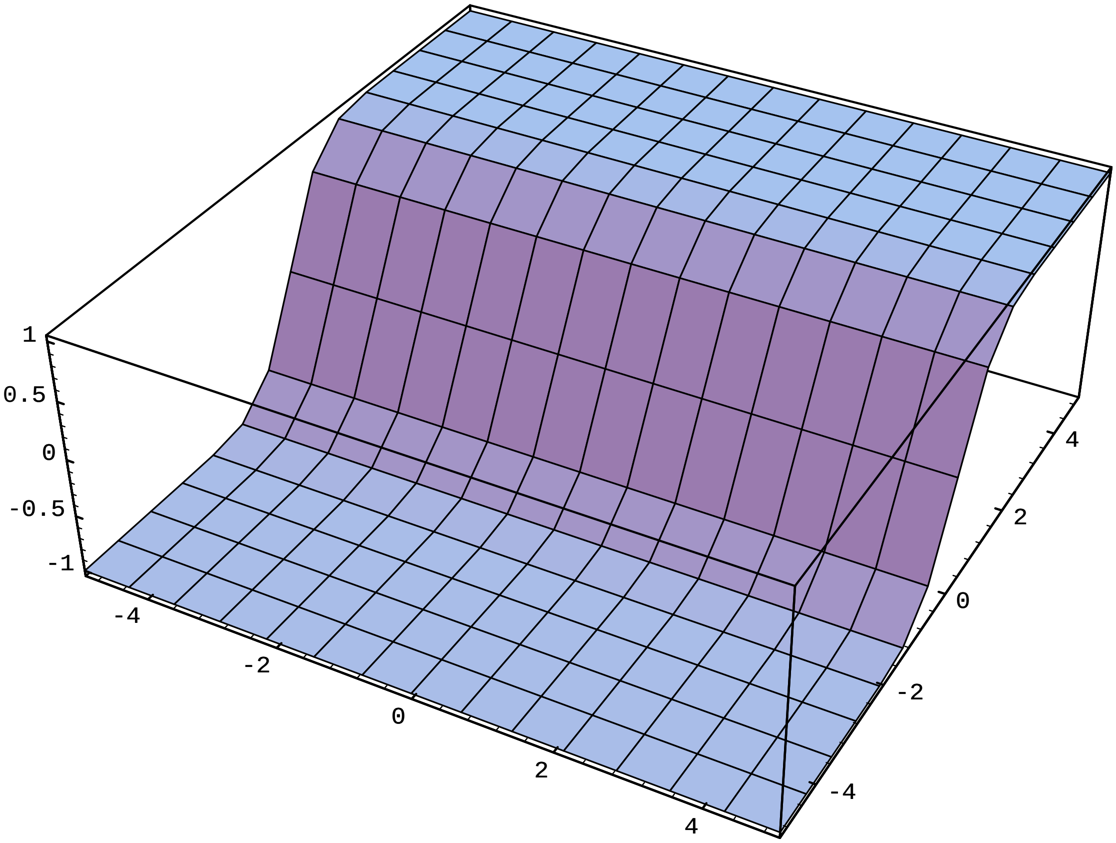

Figure 6: 'Ideal' Edge

$$
\begin{aligned}
d_x &= \frac{\partial f}{\partial x}.\frac{\partial^2 f}{\partial x^2}\cos\theta_x + \frac{\partial f}{\partial u}.\frac{\partial^2 f}{\partial u^2}\cos\theta_u + \frac{\partial f}{\partial y}.\frac{\partial^2 f}{\partial y^2}\cos\theta_y + \frac{\partial f}{\partial v}.\frac{\partial^2 f}{\partial v^2}\cos\theta_v \qquad (23)\\
d_y &= \frac{\partial f}{\partial x}.\frac{\partial^2 f}{\partial x^2}\sin\theta_x + \frac{\partial f}{\partial u}.\frac{\partial^2 f}{\partial u^2}\sin\theta_u + \frac{\partial f}{\partial y}.\frac{\partial^2 f}{\partial y^2}\sin\theta_y + \frac{\partial f}{\partial v}.\frac{\partial^2 f}{\partial v^2}\sin\theta_v \qquad (24)
\end{aligned}
$$

If we now substitute the partial derivatives of the same edge image as was used for the dilation operator (see Appendix A) into Equation 22, the response over $\varphi$ of the dilation operator can be found to be

$$
\vec{d}(x,y) = \frac{3}{4\pi}(x\cos\varphi + y\sin\varphi)e^{-2(x\cos\varphi+y\sin\varphi)^2}\begin{pmatrix}\cos\varphi\\ \sin\varphi\end{pmatrix} \qquad (25)
$$

Again this shows that the single angle returned should be an accurate indication of the orientation of the image feature.

### 3.2.3 Corner

This operator should respond to corners, or areas of high curvature. The vector returned should point *into* the corner, bisecting the angle. It should give a lower response to ordinary edge regions. It can be defined as

$$
\vec{c}(x,y) = \begin{pmatrix} c_x \\ c_y \end{pmatrix} \qquad (26)
$$

where

$$
\begin{aligned}
c_x &= \frac{\partial^2 f}{\partial x^2}.\frac{\partial f}{\partial y}\cos\theta_y + \frac{\partial f}{\partial x}.\frac{\partial^2 f}{\partial y^2}\cos\theta_x + \frac{\partial^2 f}{\partial u^2}.\frac{\partial f}{\partial v}\cos\theta_v + \frac{\partial f}{\partial u}.\frac{\partial^2 f}{\partial v^2}\cos\theta_u \qquad (27)\\
c_y &= \frac{\partial^2 f}{\partial x^2}.\frac{\partial f}{\partial y}\sin\theta_y + \frac{\partial f}{\partial x}.\frac{\partial^2 f}{\partial y^2}\sin\theta_x + \frac{\partial^2 f}{\partial u^2}.\frac{\partial f}{\partial v}\sin\theta_v + \frac{\partial f}{\partial u}.\frac{\partial^2 f}{\partial v^2}\sin\theta_u \qquad (28)
\end{aligned}
$$

As this operator responds primarily to corners rather than edges, in order to analyse its behaviour we need to define a different 'ideal' image. Instead of the edge used previously, we now use an 'ideal' corner, oriented as before at an angle $\varphi$.

$$
f(x,y) = (1 + Erf(g(x,y)))(1 + Erf(m(x,y))) \qquad (29)
$$

where

$$
\begin{aligned}
g(x,y) &= x\cos\varphi + y\sin\varphi \qquad (30)\\
m(x,y) &= -x\sin\varphi + y\cos\varphi \qquad (31)
\end{aligned}
$$

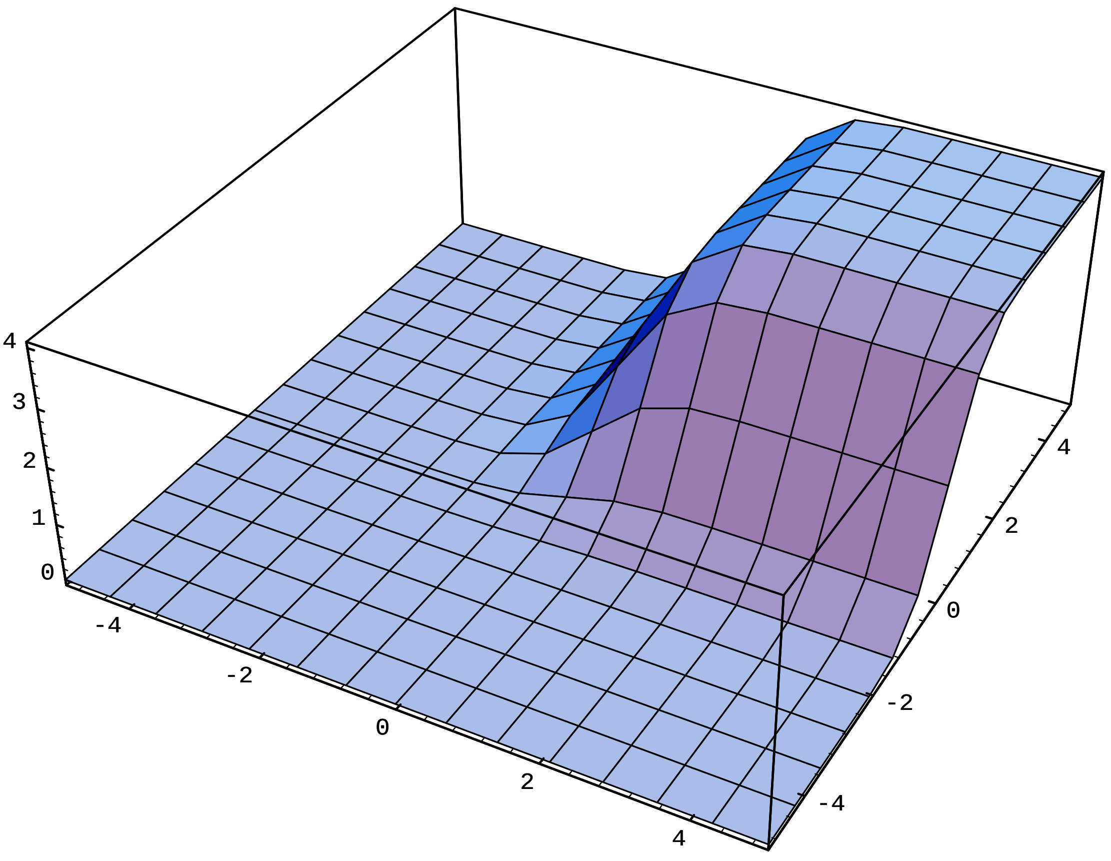

Figure 7: 'Ideal' Corner

This is illustrated in Figure 7.

Now we can calculate the partial derivatives of Equation 29 (see Appendix A) and substitute them into Equation 26 to find an expression in terms of $x$, $y$ and $\varphi$, defining the spatial variation of the response for corners of various orientations. Irrespective of the orientation of the corner however, the highest response of the operator should be at the origin of our test image, since this is where our 'ideal' corner is defined to be located. For this reason, we therefore substitute zero for both $x$ and $y$ into Equations 58-61 and 69-72, yielding the following expression for the resulting vector

$$\vec{c}(x,y) = \frac{8\sqrt{2}}{\pi\sqrt{\pi}} \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \tag{32}$$

As can be seen, the corner detector should yield an accurate vector for all orientations. This is illustrated in Figures 8 and 9.

### 3.2.4 Inside/Outside of Corner

The final feature detector returns a scalar value $s$, and gives a high response to the *insides* of corners (i.e. the acute/obtuse angle), and a low response to the *outsides* of corners (i.e. the reflex angle).

$$s = \frac{\partial^2 f}{\partial x^2}.\frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial u^2}.\frac{\partial^2 f}{\partial v^2} \tag{33}$$

## 3.3 Neuropsychological Parallels

As was mentioned in Section 3.1, Koenderink and Richards [24] have argued that two-dimensional curvature operators can be said to be more biologically plausible than are one-dimensional operators. This is because of the large amount of parallelism that is present in biological visual systems. One-dimensional operators consist of two distinct sequential processes (the calculation of an edge map, followed by differentiation of that map), whereas two-dimensional operators which operate directly on the luminance image have a greater degree of inherent parallelism. This would seem to make them a more plausible model for the detection of curvature in mammalian visual systems.

## 3.4 Computational Requirements

The basic computational cost of the two initial partial derivative filters (the $u$ and $v$ axes) will obviously be 9 multiplications for each filter. The second derivatives of each require the appropriate filter to be applied again, and so each cost another 9 multiplications. In addition to this, we also need to calculate $\frac{\partial^2 f}{\partial uv}$, which will take another 9 multiplications. This gives a total of 45 multiplications for the derivatives in the $u$ and $v$ axes. Combining the values of these to determine the first derivatives in the $x$

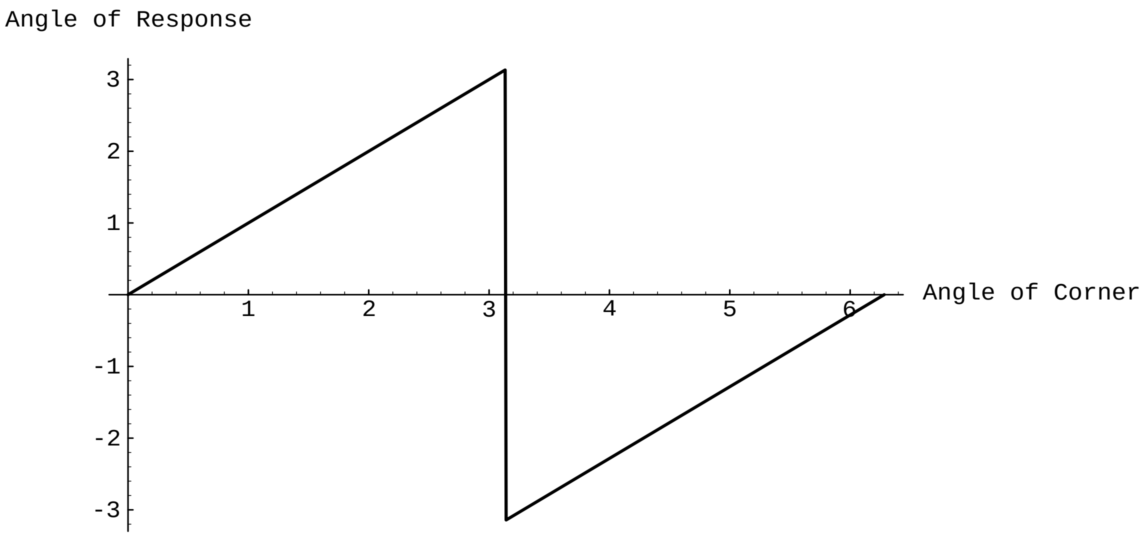

Figure 8: Graph Showing Vector Argument vs Orientation of Corner

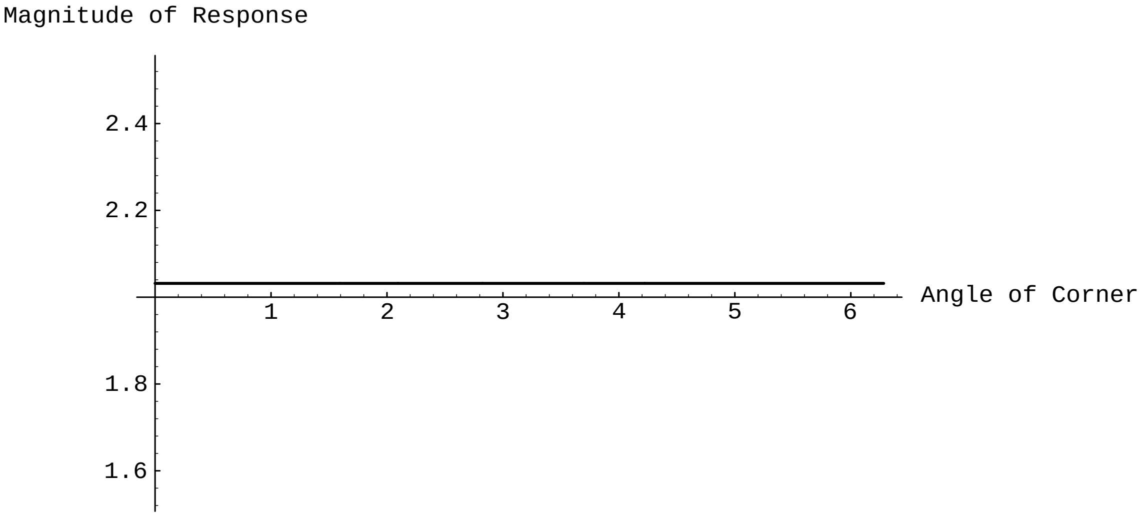

Figure 9: Graph Showing Vector Magnitude vs Orientation of Corner

| *Process* | *No. Multiplications* |
|---|---:|
| Translation | 2 |
| Dilation | 16 |
| Corner | 16 |
| Inside/Outside of Corner | 2 |

Table 3: The Computational Cost (Per Image Pixel) of the Feature Detection Filters

and $y$ directions according to Equations 12 and 13, costs another 2 multiplications. Similarly the second partial derivatives in $x$ and $y$ according to Equations 14 and 15, will cost another 2 multiplications. Thus we have a final total of 49 multiplications for the calculation of all of the partial derivatives.

The additional requirements of the combinations of the results of these derivative filters are shown in Table 3, and represent a relatively small overhead compared with the basic filtering operation.

## 3.5 Results

A greyscale representation of the result of applying the orientation, dilation, and corner detection filters on a clean version of the shapes image (see Figure 10) are shown in Figures 11, 12, and 13.

An analysis of the behaviour of the corner detection operator is shown in Figure 15. This illustrates that the angle of the vector returned by the operator gives a reasonably accurate indication of the orientation of the corner in the image. The results were obtained by applying the operator to a succession of 90 degree corners at varying orientations.

The result of applying the inside/outside of corners filter is illustrated in Figure 14.

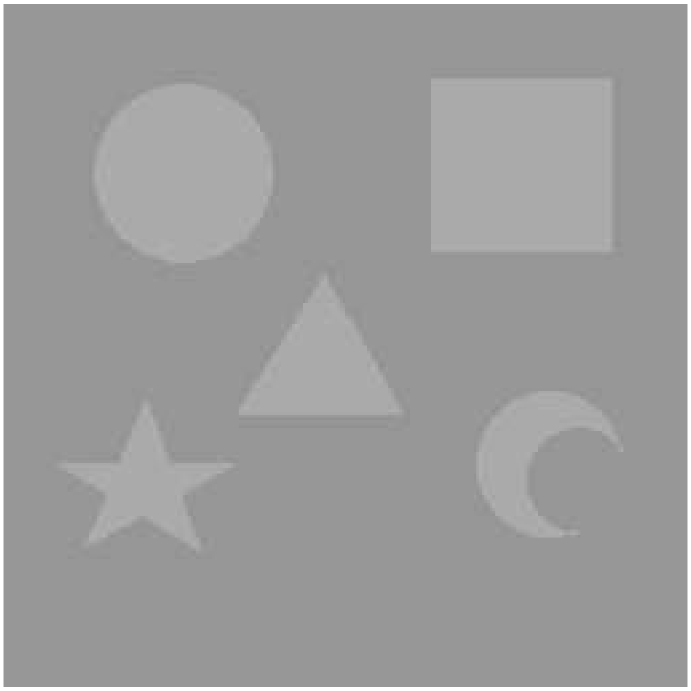

Figure 10: Clean Shapes Image

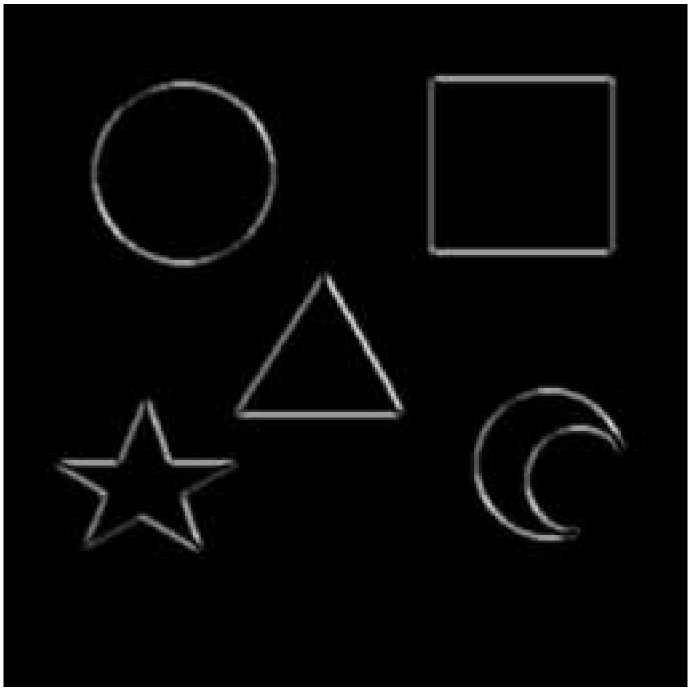

Figure 11: Results of the Translation Operator on the Shapes Image

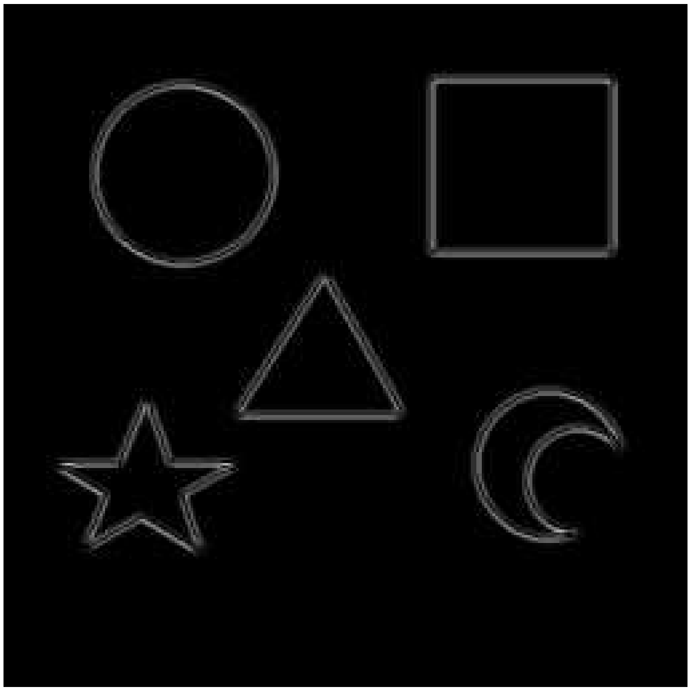

Figure 12: Results of the Dilation Operator on the Shapes Image

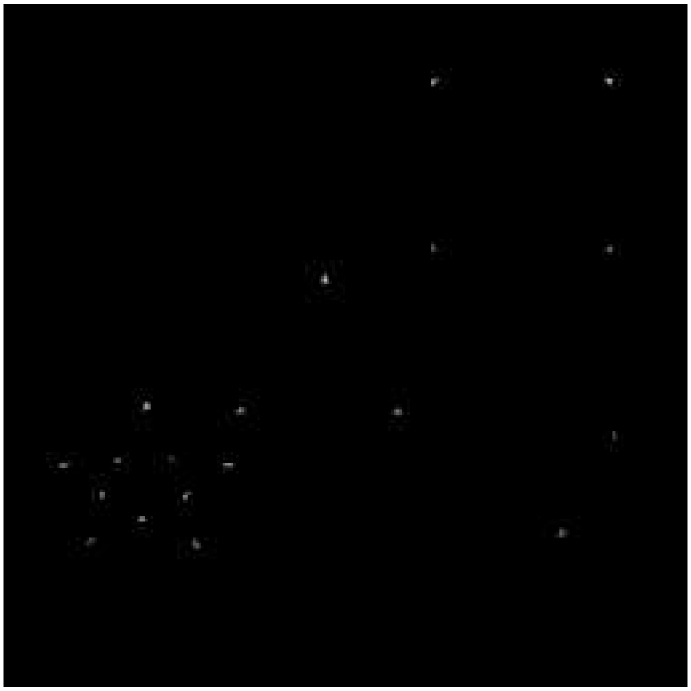

Figure 13: Results of the Corner Operator on the Shapes Image

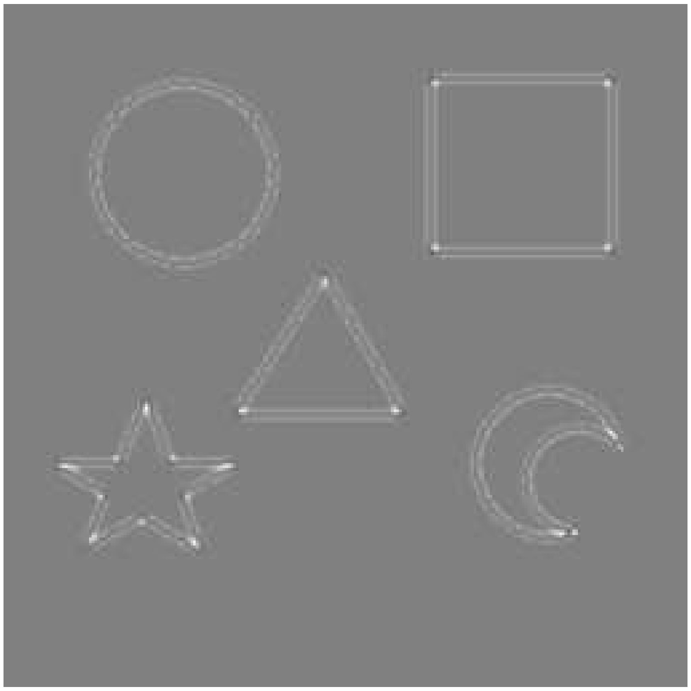

Figure 14: Results of Inside/Outside Operator on the Shapes Image

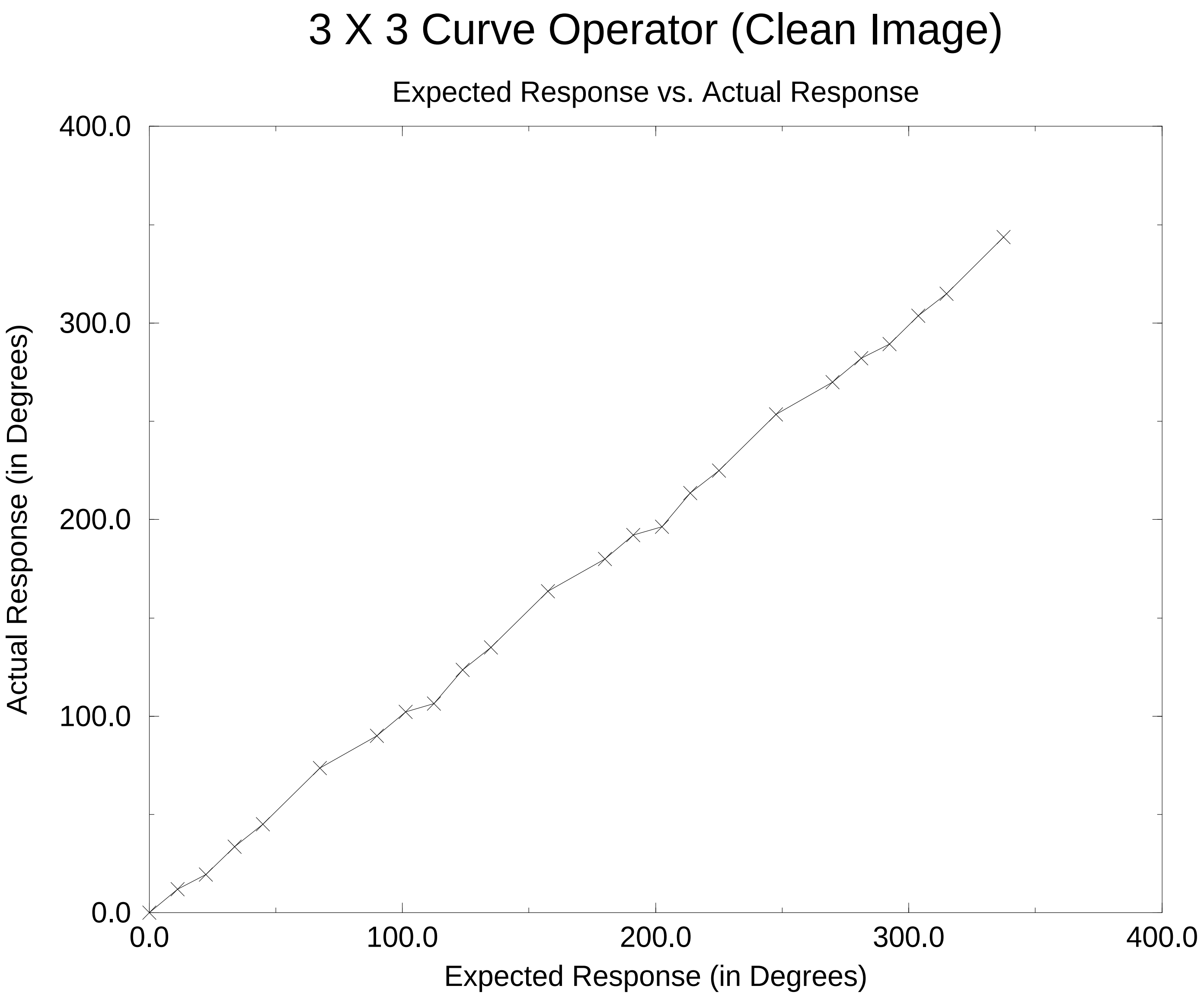

Figure 15: Graph Showing the Performance of the Corner Operator

# 4 Conclusions And Future Work

## 4.1 Summary

This report has described two separate lines of work. Firstly, there was the image restoration algorithm. This has produced encouraging results on artificial images, although its performance on natural images is not as impressive. Secondly, there was the feature extraction filters. Work is still proceeding in this area, and results so far are again encouraging. These two seemingly distinct areas of work are united by the common theme of the role of local symmetries in visual processing. A review has been given of the fields of fractal coding [20] and Lie algebra ([16], [11]), both of which deal with this theme. This is not a new concept then, although it would appear that for such a promising field it has been comparatively neglected. It would seem almost certain that the role of transformations in a model of mammalian vision is pivotal. In particular, the three basic pairs of transformations suggested by the LTG model of visual processing would seem to be especially significant.

## 4.2 Further Work on Image Restoration

One possible way of making an improvement to the results of the restoration algorithm could be to utilise some information from the results of the feature extraction process. Currently, after the multiresolution smoothing, the edge regions are sharpened by contracting the image towards the notional edge. This is effectively applying a transformation which is a symmetry when applied at the clean edge. However, at corner regions, a different symmetry will be required to sharpen the image, and it is possible that the information derived from the feature extraction filters could be used to decide which transformation to use, and hopefully make an improvement in results.

## 4.3 Further Work on Feature Extraction

As was stated above, work on the feature extraction algorithm is still ongoing. In particular, the filters' responses at differing scales have not been examined in any great depth as yet. Also, it may be possible to improve the accuracy of the information returned by the filters by combining the information from several different filters.

## 4.4 Unification - Towards a Model of Object Recognition

The scope of the local image symmetry concept is not limited to just restoration and feature extraction. Other areas of image processing could benefit from considering the implications and possibilities of the role of symmetries in vision. The future direction of this work will bear this in mind. It is envisaged, that the current work will progress with the aim of an eventual unification of the two areas so far examined, whilst opening up new avenues to explore. The ultimate aim will be to work towards a model of object recognition, based on the analysis of the local image symmetries of the object. It is likely that this model will incorporate some kind of connectionist approach to the problem. One possibility here would be the introduction of a Hopfield Net to recognize information concerning local image symmetries derived from the feature extraction process.

# A Calculation of the Partial Derivatives

This appendix shows the calculation of the first and second order partial derivatives of two 'ideal' images, an ideal edge and an ideal corner. Each of these is dealt with seperately below. These derivatives are used in the analysis of the behaviour of three of the feature extraction operators described in Section 3.2.

## A.1 Edge Image

The ideal edge image is illustrated in Figure 6. It is oriented at an angle $\varphi$ and it is defined by

$$f(x,y) = Erf(g(x,y)) \tag{34}$$

where

$$g(x,y) = x\cos\varphi + y\sin\varphi \tag{35}$$

The defining equation of this edge image can now be differentiated to yield the first and second order partial derivatives with respect to $x$ and $y$.

$$\begin{aligned}
\frac{\partial f}{\partial x} &= \frac{2}{\sqrt{\pi}}\cos\varphi.e^{-g(x,y)^2} &(36)\\
\frac{\partial^2 f}{\partial x^2} &= \frac{-4}{\sqrt{\pi}}\cos^2\varphi.g(x,y).e^{-g(x,y)^2} &(37)\\
\frac{\partial f}{\partial y} &= \frac{2}{\sqrt{\pi}}\sin\varphi.e^{-g(x,y)^2} &(38)\\
\frac{\partial^2 f}{\partial y^2} &= \frac{-4}{\sqrt{\pi}}\sin^2\varphi.g(x,y).e^{-g(x,y)^2} &(39)
\end{aligned}$$

To find the partial derivatives with respect to $u$ and $v$, we need to rotate the axes by $\pi/4$ radians. This is done using the following relationships

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \tag{40}$$

therefore

$$\begin{aligned}
x &= \frac{u+v}{\sqrt{2}} &(41)\\
y &= \frac{u-v}{\sqrt{2}} &(42)
\end{aligned}$$

$$u = \frac{x+y}{\sqrt{2}} \tag{43}$$

$$v = \frac{x-y}{\sqrt{2}} \tag{44}$$

Now from Equations 41 and 42, we can express the edge image (Equation 18) in terms of $u$ and $v$ as follows.

$$f(u,v) = Erf(h(u,v)) \tag{45}$$

where

$$h(u,v) = \frac{1}{\sqrt{2}}(u(\cos\varphi + \sin\varphi) + v(\cos\varphi - \sin\varphi)) \tag{46}$$

The partial derivatives with respect to $u$ and $v$ are then equal to

$$\frac{\partial f}{\partial u} = (\sqrt{\frac{2}{\pi}})(\cos\varphi + \sin\varphi).e^{-h(u,v)^2} \tag{47}$$

$$\frac{\partial^2 f}{\partial u^2} = -(\sqrt{\frac{2}{\pi}})(\cos\varphi + \sin\varphi)^2.(u(\cos\varphi + \sin\varphi) + v(\cos\varphi - \sin\varphi))e^{-h(u,v)^2} \tag{48}$$

$$\frac{\partial f}{\partial v} = (\sqrt{\frac{2}{\pi}})(\cos\varphi - \sin\varphi).e^{-h(u,v)^2} \tag{49}$$

$$\frac{\partial^2 f}{\partial v^2} = -(\sqrt{\frac{2}{\pi}})(\cos\varphi - \sin\varphi)^2.(u(\cos\varphi + \sin\varphi) + v(\cos\varphi - \sin\varphi))e^{-h(u,v)^2} \tag{50}$$

Using the relationships between $x$ and $y$, and $u$ and $v$ (Equations 43 and 44), we can now express these derivatives in terms of just $x$ and $y$.

$$\frac{\partial f}{\partial u} = (\sqrt{\frac{2}{\pi}})(\cos\varphi + \sin\varphi).e^{-g(x,y)^2} \tag{51}$$

$$\frac{\partial^2 f}{\partial u^2} = -(\frac{2}{\sqrt{\pi}})(\cos\varphi + \sin\varphi)^2.(x\cos\varphi + y\sin\varphi)e^{-g(x,y)^2} \tag{52}$$

$$\frac{\partial f}{\partial v} = (\sqrt{\frac{2}{\pi}})(\cos\varphi - \sin\varphi).e^{-g(x,y)^2} \tag{53}$$

$$\frac{\partial^2 f}{\partial v^2} = -(\frac{2}{\sqrt{\pi}})(\cos\varphi - \sin\varphi)^2.(x\cos\varphi + y\sin\varphi)e^{-g(x,y)^2} \tag{54}$$

## A.2 Corner Image

The ideal corner image is illustrated in Figure 7. It is oriented at an angle $\varphi$ and it is defined by

$$f(x,y) = (1 + Erf(g(x,y)))(1 + Erf(m(x,y))) \tag{55}$$

where

$$g(x,y) \;=\; x\cos\varphi + y\sin\varphi \tag{56}$$

$$m(x,y) \;=\; -x\sin\varphi + y\cos\varphi \tag{57}$$

This equation can be differentiated to yield the first and second order partial derivatives with respect to $x$ and $y$.

$$\begin{aligned}\frac{\partial f}{\partial x} = (\tfrac{2}{\sqrt{\pi}})( \quad & \cos\varphi(1 + Erf(m(x,y))).e^{-g(x,y)^2} - \\ & \sin\varphi(1 + Erf(g(x,y))).e^{-m(x,y)^2})\end{aligned} \tag{58}$$

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} = -(\tfrac{4}{\sqrt{\pi}})( \quad & \frac{2}{\sqrt{\pi}}\cos\varphi\sin\varphi.e^{-g(x,y)^2 - m(x,y)^2} + \\ & m(x,y)\sin^2\varphi(1 + Erf(g(x,y))).e^{-m(x,y)^2} + \\ & g(x,y)\cos^2\varphi(1 + Erf(m(x,y))).e^{-g(x,y)^2})\end{aligned} \tag{59}$$

$$\begin{aligned}\frac{\partial f}{\partial y} = (\tfrac{2}{\sqrt{\pi}})( \quad & \cos\varphi(1 + Erf(g(x,y))).e^{-m(x,y)^2} + \\ & \sin\varphi(1 + Erf(m(x,y))).e^{-g(x,y)^2})\end{aligned} \tag{60}$$

$$\begin{aligned}\frac{\partial^2 f}{\partial y^2} = -(\tfrac{4}{\sqrt{\pi}})( \quad & -\frac{2}{\sqrt{\pi}}\cos\varphi\sin\varphi.e^{-g(x,y)^2 - m(x,y)^2} + \\ & g(x,y)\sin^2\varphi(1 + Erf(m(x,y))).e^{-g(x,y)^2} + \\ & m(x,y)\cos^2\varphi(1 + Erf(g(x,y))).e^{-m(x,y)^2})\end{aligned} \tag{61}$$

As before we can now determine the partial derivatives with respect to $u$ and $v$ by substituting the values for $x$ and $y$ according to Equations 41 and 42 into Equation 29.

$$f(u,v) = (1 + Erf(h(u,v)))(1 + Erf(n(u,v))) \tag{62}$$

where

$$h(u,v) = \frac{1}{\sqrt{2}}(u(\cos\varphi + \sin\varphi) + v(\cos\varphi - \sin\varphi)) \tag{63}$$

$$n(u,v) = \frac{1}{\sqrt{2}}(u(\cos\varphi - \sin\varphi) - v(\cos\varphi + \sin\varphi)) \tag{64}$$

The partial derivatives with respect to $u$ and $v$ can then be expressed as

$$
\begin{aligned}
\frac{\partial f}{\partial u} &= (\sqrt{\tfrac{2}{\pi}})( && (\cos\varphi - \sin\varphi)(1 + Erf(h(u,v))).e^{-n(u,v)^2} + \\
& && (\cos\varphi + \sin\varphi)(1 + Erf(n(u,v))).e^{-h(u,v)^2}) \qquad (65) \\
\frac{\partial^2 f}{\partial u^2} &= -(\tfrac{2}{\sqrt{\pi}})( && -\frac{2}{\sqrt{\pi}}(\cos\varphi - \sin\varphi)(\cos\varphi + \sin\varphi).e^{-h(u,v)^2 - n(u,v)^2} + \\
& && (\cos\varphi + \sin\varphi)^2 h(u,v)(1 + Erf(n(u,v))).e^{-h(u,v)^2} + \\
& && (\cos\varphi - \sin\varphi)^2 n(u,v)(1 + Erf(h(u,v))).e^{-n(u,v)^2}) \qquad (66) \\
\frac{\partial f}{\partial v} &= (\sqrt{\tfrac{2}{\pi}})( && (-\cos\varphi - \sin\varphi)(1 + Erf(h(u,v))).e^{-n(u,v)^2} + \\
& && (\cos\varphi - \sin\varphi)(1 + Erf(n(u,v))).e^{-h(u,v)^2}) \qquad (67) \\
\frac{\partial^2 f}{\partial v^2} &= -(\tfrac{2}{\sqrt{\pi}})( && -\frac{2}{\sqrt{\pi}}(-\cos\varphi - \sin\varphi)(\cos\varphi - \sin\varphi).e^{-h(u,v)^2 - n(u,v)^2} + \\
& && (\cos\varphi - \sin\varphi)^2 h(u,v)(1 + Erf(n(u,v))).e^{-h(u,v)^2} + \\
& && (-\cos\varphi - \sin\varphi)^2 n(u,v)(1 + Erf(h(u,v))).e^{-n(u,v)^2}) \qquad (68)
\end{aligned}
$$

Again, using the relationship between $x$ and $y$, and $u$ and $v$ (Equations 43 and 44), we can now express these derivatives in terms of just $x$ and $y$.

$$
\begin{aligned}
\frac{\partial f}{\partial u} &= (\sqrt{\tfrac{2}{\pi}})( && (\cos\varphi - \sin\varphi)(1 + Erf(g(x,y))).e^{-m(x,y)^2} + \\
& && (\cos\varphi + \sin\varphi)(1 + Erf(m(x,y))).e^{-g(x,y)^2}) \qquad (69) \\
\frac{\partial^2 f}{\partial u^2} &= -(\tfrac{2}{\sqrt{\pi}})( && -\frac{2}{\sqrt{\pi}}(\cos\varphi - \sin\varphi)(\cos\varphi + \sin\varphi).e^{-g(x,y)^2 - m(x,y)^2} + \\
& && \frac{1}{\sqrt{2}}((\cos\varphi + \sin\varphi)^2 g(x,y)(1 + Erf(m(x,y))).e^{-g(x,y)^2} + \\
& && \frac{1}{\sqrt{2}}((\cos\varphi - \sin\varphi)^2 m(x,y)(1 + Erf(g(x,y))).e^{-m(x,y)^2}) \qquad (70) \\
\frac{\partial f}{\partial v} &= (\sqrt{\tfrac{2}{\pi}})( && (\cos\varphi - \sin\varphi)(1 + Erf(m(x,y))).e^{-g(x,y)^2} - \\
& && (\cos\varphi + \sin\varphi)(1 + Erf(g(x,y))).e^{-m(x,y)^2}) \qquad (71) \\
\frac{\partial^2 f}{\partial v^2} &= -(\tfrac{2}{\sqrt{\pi}})( && -\frac{2}{\sqrt{\pi}}(-\cos\varphi - \sin\varphi)(\cos\varphi - \sin\varphi).e^{-g(x,y)^2 - m(x,y)^2} + \\
& && \frac{1}{\sqrt{2}}(\cos\varphi - \sin\varphi)^2 g(x,y)(1 + Erf(m(x,y))).e^{-g(x,y)^2} + \\
& && \frac{1}{\sqrt{2}}(-\cos\varphi - \sin\varphi)^2 m(x,y)(1 + Erf(g(x,y))).e^{-m(x,y)^2}) \qquad (72)
\end{aligned}
$$

# References

[1] M. T. Anderson. *Controllable Multidimensional Filters and Models in Low Level Computer Vision.* PhD thesis, Department of Electrical Engineering, Linköping University, Sweden, 1992.

[2] M. Barnsley. *Fractals Everywhere.* Academic Press, 1988.

[3] A. H. Bhalerao. *Multiresolution Image Segmentation.* PhD thesis, Department of Computer Science, The University of Warwick, UK, October 1991.

[4] J. Bigün. *Local Symmetry Features in Image Processing.* PhD thesis, University of Linköping, Sweden, 1988.

[5] W. F. Bischof and T. Caelli. Parsing Scale-Space and Spatial Stability Analysis. *Computer Vision, Graphics, and Image Processing*, 42:192–205, 1988.

[6] M. Brady and H. Asada. Smooth and Local Symmetries and their Implementation. *International Journal of Robotics*, 3:36–61, 1984.

[7] J. M. H. Du Buf and T. G. Campbell. A Quantitative Comparison of Edge-Preserving Smoothing Techniques. *Signal Processing*, 21:289–301, 1990.

[8] P. J. Burt and E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communication*, COM-31:532–540, 1983.

[9] S. Clippingdale. *Multiresolution Image Modelling and Estimation.* PhD thesis, Department of Computer Science, The University of Warwick, UK, September 1988.

[10] I. Daubechies. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory*, 36:961–1004, 1990.

[11] P. C. Dodwell. The Lie Transformation Group Model of Visual Perception. *Perception and Psychophysics*, 34:1–16, 1983.

[12] W. T. Freeman. *Steerable Filters and Local Analysis of Image Structure.* PhD thesis, Massachusetts Institute of Technology, USA, 1992.

[13] J. J. Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, 1979.

[14] G. H. Granlund. In Search of a General Picture Processing Operator. *Computer Graphics and Image Processing*, 8:155–173, 1978.

[15] E. Heinänen, A. Nieminen, P. Heinonen, and Y. Neuvo. FIR-Median Hybrid Filters for Image Processing. In *Proc. 3rd European Signal Processing Conference*, 1986.

[16] W. C. Hoffman. The Neuron as a Lie Group Germ and a Lie Product. *Quarterly Journal of Applied Mathematics*, 25:423–440, 1965.

[17] D. H. Hubel. *Eye, Brain, and Vision.* Scientific American Library, 1988.

[18] D. H. Hubel and T. N. Wiesel. Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex. *Journal of Physiology*, 160:106–154, 1962.

[19] D. H. Hubel and T. N. Wiesel. Functional Architecture of Macaque Monkey Striate Cortex. *Proceedings of the Royal Society (Series B)*, 198:1–59, 1977.

[20] A. E. Jacquin. A Novel Fractal Block-Coding Technique for Digital Images. In *Proc. ICASSP '90*, 1990.

[21] M. Knoll, J. Kugler, J. Eichmeier, and O. Höfer. Note on the Spectroscopy of Subjective Light Patterns. *Journal of Analytical Psychology*, 7, 1962.

[22] H. Knutsson. *Filtering and Reconstruction in Image Processing.* PhD thesis, University of Linköping, Sweden, 1982.

[23] J. J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50:363–370, 1984.

[24] J. J. Koenderink and W. Richards. Two-Dimensional Curvature Operators. *Journal of the Optical Society of America*, 5:1136–1141, 1988.

[25] J. J. Koenderink and A. J. van Doorn. Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55:367–375, 1987.

[26] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Affine Invariant Model-Based Object Recognition. *IEEE Transactions on Robotics and Automation*, 6:578–588, 1990.

[27] P. Lövsund. *Biological Effects of Alternating Magnetic Fields with Special Reference to the Visual System.* PhD thesis, Linköping University, Sweden, 1980.

[28] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag, 1990.

[29] A. Rattarangsi and R. T. Chin. Scale-Based Detection of Corners of Planar Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:430–449, 1992.

[30] W. Richards, B. Dawson, and D. Whittington. Encoding Contour Shape by Curvature Extrema. *Journal of the Optical Society of America*, 3:1483–1491, 1986.

[31] P. Simard, Y. Le Cun, B. Victorri, and J. Denker. Tangent Prop - A Formalism for Specifying Selected Invariances in an Adaptive Network. In *Proc. 5th Conference on Neural Information Processing Systems*, 1991.

[32] M. R. Teague. Image Analysis via the General Theory of Moments. *Journal of the Optical Society of America*, 70:920–930, 1980.

[33] R. Wilson. Quadtree Predictive Coding: A New Class of Image Data Compression Algorithms. In *Proc. ICASSP '84, San Diego, CA*, 1984.

[34] R. Wilson and A. H. Bhalerao. Kernel Designs for Efficient Multiresolution Edge Detection and Orientation Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:384–390, 1992.

[35] A. P. Witkin. Scale-Space Filtering. In *Proc. Int. Joint Conference on Artificial Intelligence*, 1983.