Object Representation Using Circular Harmonics

Nicola Cross, Roland Wilson
Department of Computer Science,
University of Warwick,
Coventry

August 2, 1994

1 Abstract

This report is separated into two parts. In the first part a review of object representations is image analysis is presented. In the second part a new representation is proposed. The system presented here is one that can be used to project information describing the corners of an object from some storage area, down onto a network. The idea is to reinforce the detection of object boundaries performed by a Hopfield network. These networks are efficient in terms of detecting edges, but are less efficient at detecting corners. This is because the Hopfield network works by assuming that an edge will continue in a more or less straight line. A *ripple filter* is used; this projects ripples of vectors onto the areas where corners are expected. Each vector consists of: a magnitude that indicates a level of certainty about the position of the corner, and a direction which indicates the expected orientation of the corner. The vectors are used by the Hopfield network to override existing straight links with curved links, so helping it to detect corners. We describe a method for projecting sectors of the ripples using circular harmonics. Possible extensions and improvements to the work are considered. In **??** a possible network architecture for a *what-where* vision system that can handle multiple, duplicate Whats is presented. A suitable object representation for this network is suggested.

Contents

List of Figures

2 Introduction

2.1 Object Representation

All vision systems that perform object recognition have in common the need to make comparisons between what has been detected and some stored representation of objects. Many schemes for object representation have been used. Hummel and Beiderman [6] for example, used a system of geons (simple 3-dimensional geometric shapes); each object representation consists of a set of geons and their positional relations to each other. Still others have been based on the generalised cylinder; the cylinder might be bashed, stretched and squeezed in a number of mathematical ways to fit a 3D object description. Approaches based on geons, cylinders and the like are more suited to man made environments such as the factory conveyor belt, or artificial worlds like 'block' worlds worked on by many including Waltz [9] than to the modelling of natural classes of objects like mountain ranges. Fractal modelling can be used to create the most realistic looking models of natural objects as Mandelbrot's work shows [7]; and fractal descriptions can be quite concise. Another alternative is to store the entire image. Clearly this is costly in terms of processing time and storage space, especially where multiple views of the same object must be stored.

2.2 Object Representation by Corners

The method presented here uses a compact representation. All information about the image except for that pertaining to the corners is discarded. What remains gives an object description in terms of the positions of the corners and their orientations. Information for one corner comprises: spatial information as an x, y coordinate, and two values which give the limits of an angle range.

2.3 The Hopfield Net

A modified Hopfield network is used to perform edge detection on images. The Hopfield net performs well in terms of detecting straight and gently curving edges. Corners are also detected on clean images, but when noise is injected into the image, corner detection suffers most. This is because the Hopfield net works by assuming that an edge will continue in a more or less straight line. The stored corner information is useful because it can be used to override links in the network, 'helping' the Hopfield net to turn corners. The net result in noisy images is that both the corners and edges of objects are more complete. Some examples of this are shown in section 4.

2.4 Psychological Relevance

This method has some intuitive appeal in that it has similarities to visual illusions such as the Kanisza triangle shown in Figure 1. The same Gestalt idea of closure is evident if the corner information is seen as the 'top down' reinforcement of the 'bottom up' illusion of the three non-existent straight lines that form the illusory triangle.

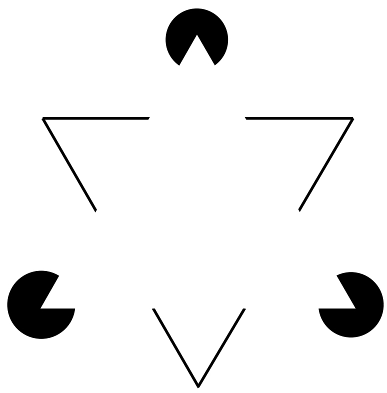

Figure 1: The Kanisza Triangle

The Gestalt psychologists were much concerned with 'perceptual organisation', seeing the process of visual perception as being largely concerned with organising stimulus patterns into wholes (Gestalten). This organising into wholes was demonstrated with black and white figures, mainly patterns concerned with dots. As Gregory tells us [4], even random dot patterns tend to be organised into configurations. Gestalt ideas have become unfashionable, mostly because they saw these tendencies as being innate and exclusively 'top down' processes, and partly because their theories were seen as 'non explanatory'. It is nevertheless undeniable that grouping of visual stimuli does occur according to proximity, similarity, and 'common fate' - the related movements of the different parts of the same object that make it appear as a whole. This need not be seen as an innate ability; it might as easily be seen as learned. Current knowledge about the human visual system also tells us that a large amount of preprocessing of visual information occurs before it reaches the 'higher' centres of the brain - see Hubel [5]. But here again, Gestalt ideas can still be seen as valid within the contemporary view of human visual processing as an interactive one between 'bottom up' and 'top down' processes.

2.5 Circular Harmonics

The major part of this work is involved not with the details of the representation, but with a method for projecting the stored details down onto a layer within a network.

This is done by using filters which are designed to cause a *ripple* in the output at each corner point. See Figure 2 for an illustration of the full ripple. The ripples consist of concentric rings of vectors which always point outwards from the origin of the circle and towards the circumference. Vectors are strongest near the origin of the circles, becoming progressively weaker as the distance from the origin increases. The strength of the vector indicates how certain it is that the corner point will be found in this position. The length of the arrows in Figure 2 reflect the strength of the vectors.

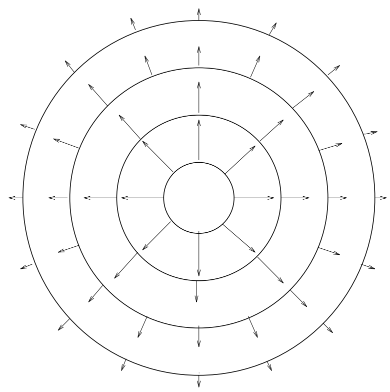

Figure 2: A Complete Ripple

The complete ripple is never output. It is possible to output just a sector of the ripple; this then indicates the orientation of the corner. Vectors always point towards the insides of corners. Note that a 'corner' could be a steep curve, a simple 2-dimensional corner or a vertex. The more *acute* the corner, the *smaller* the specified sector of ripple will be. Partial ripples are shown in Figure 3.

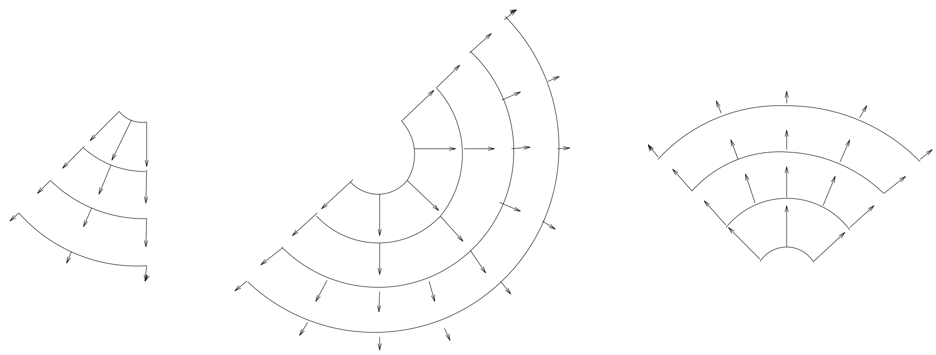

Figure 3: Partial Ripples

The mathematical method used to achieve partial output of the ripple involves the use of circular harmonics. If a complete ripple were always output then just one filter would be required. However, if the filter output is set to be the sum of a series of filters instead, then each of the filters in the series can be manipulated so that when the series is added together only the required section of the ripple is output. Each filter becomes a member of a circular harmonic series. The mathematics underlying the circular harmonics is described in section 3.

2.6 Quadtrees

A quadtree structure is used for mapping the two levels of the network. The input level is one step further up the pyramid than the output level (the ripple layer). Each parent node in the input layer thus has four children in the output layer, and each child node in the output layer has just one parent node in the input layer. Activation always flows from the input layer to the output layer. The architecture is illustrated in Figure 4. The ripple filtering system thus works as a pyramidal feed-forward network. No learning is involved; the weights are predetermined by a filter and the corner information. The pyramid consists of only two levels and so is very small, but in section 6 the possibility of implementing the circular harmonics within a multi-resolution framework is discussed.

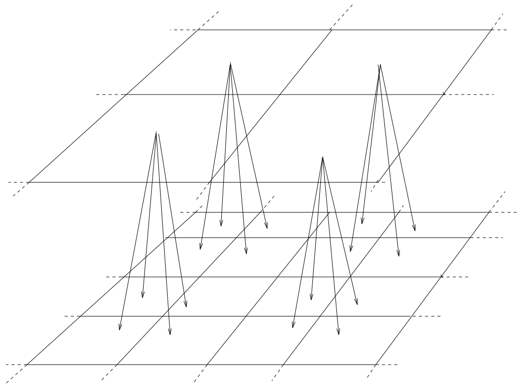

Figure 4: Quad Tree Structure

Burt and Adelson [2] used multiresolution pyramids to build their Gaussian and Laplacian pyramids. Although these did not have a quadtree construction, they did use a gaussian weighting function . Bhalerao [1] used quadtrees for multiresolution image segmentation; boundaries and regions are estimated, then iteratively improved. Coarse features such as large regions are detailed higher up the pyramid than fine features such as boundaries, which are detailed at the lower levels. Clippingdale [3] used the multiresolution pyramid to restore clean images from noisy ones. As another example, Wilson [10] used quadtree pyramids for predictive image coding. The use of multiresolution pyramids for computer vision systems is intuitively appealing because it provides the framework for the analysis of objects at several different scales.

Humans perform scale invariance in a seemingly effortless manner, and we know that there are complex cells in the retina that respond to features at different scales. There is above all, a need for visual systems to compress and coalesce image data without losing detail. The multiresolution pyramid provides a ideal way of doing this.

3 Theory

The original filter is an NxN array. Each element of the array contains a complex number. In polar form, the argument of each complex number gives a direction; this is calculated from the angle between the filter element and the positive direction of the x axis. The modulus takes the value of the distance from the centre of the filter, modified by a Gaussian function. More formally:

$$\angle w(n,m) = e^{j\angle(x,y)} = e^{j\theta} \tag{1}$$

and

$$|w(n,m)| = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

Where: n and m are the filter indices, x and y are the distances of the filter element $w(n,m)$ in the x and y directions relative to the centre of the filter, σ^2 represents the variance, and θ is the angle whose (signed) tangent is the ratio between x and y.

Figure 5 shows how the Gaussian function acts to emphasise the moduli of the filter values closest to the centre of the filter, and to diminish those furthest away. σ is chosen to give the Gaussian function a value of near 1 at the centre of the filter, and near zero at the edges of the filter. Figure 6 illustrates the directions given by the arguments of the complex filter elements. This fundamental unit becomes the first harmonic of a series. In the next step, the filter is expanded into a series by

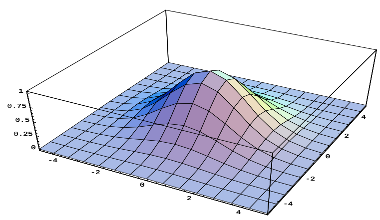

Figure 5: The Gaussian Function

expanding the function $e^{j\theta}$. The Gaussian function is ignored for the moment. There is a series such that:

$$f(\theta) = e^{j\theta} = \sum_i Ci e^{ji\theta} \tag{3}$$

ie:

$$f(\theta) = e^{j\theta} = \cdots + C_{-2}e^{-j2\theta} + C_{-1}e^{-j\theta} + C_0 + C_1 e^{j\theta} + C_2 e^{j2\theta} + \cdots \tag{4}$$

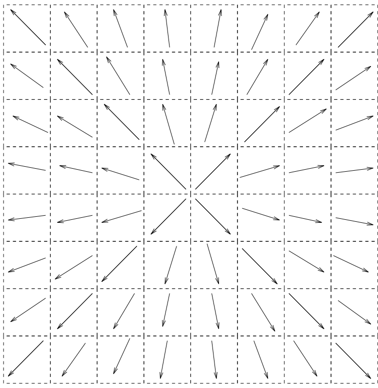

Figure 6: The Arguments in the Complex Filter

To find the coefficient for a given harmonic (ie: a given value of i) over a specified range of θ, from θ_1 to θ_2 say, the function $e^{j\theta}$ is integrated within these limits - see figure 7: The function:

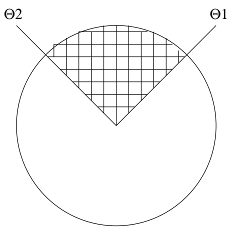

Figure 7: Integrating over a specified range

$$f(\theta) = \sum_i Cie^{ji\theta} \tag{5}$$

becomes:

$$f(\theta - \psi) = \sum_i C_i e^{ji(\theta-\psi)} = \sum_i C_i' \, e^{ji\theta}, \quad C_i' = C_i e^{-ji\psi} \tag{6}$$

because of the trigonometric identity:

$$\cos(\theta - \psi) = \cos\theta\cos\psi + \sin\theta\sin\psi \tag{7}$$

Fourier coefficients are calculated:

$$C_i = \frac{1}{2\pi}\int_{\theta_1}^{\theta_2} e^{j\theta} \, e^{-ji\theta} \, d\theta, \quad \theta_1 \le \theta < \theta_2 \tag{8}$$

$$= \frac{1}{2\pi}\int_{\theta_1}^{\theta_2} e^{j(i-1)\theta} \, d\theta \tag{9}$$

$$= \frac{1}{2\pi(-j(i-1))}\left(e^{-j(i-1)\theta_2} - e^{-j(i-1)\theta_1}\right) \tag{10}$$

$$= \frac{e^{-j(i-1)\frac{\theta_1+\theta_2}{2}}\left(e^{j(i-1)\frac{\theta_1-\theta_2}{2}} - e^{-j(i-1)\frac{\theta_1-\theta_2}{2}}\right)}{-2\pi j(i-1)} \tag{11}$$

$$= \frac{e^{-j(i-1)\frac{\theta_1+\theta_2}{2}}\, 2j \sin\left(i-1\right)\frac{\theta_1-\theta_2}{2}}{-2\pi j(i-1)} \tag{12}$$

$$= -e^{-j(i-1)\frac{\theta_1+\theta_2}{2}}\,\frac{\sin(i-1)\frac{\theta_1-\theta_2}{2}}{\pi(i-1)} \tag{13}$$

In the case where $i = 1$ the sine function is tending to zero and so is the term $(i-1)\pi$. L'Hôpital's rule for *finding the limit of a ratio of two functions each of which separately tends to zero* is used. It states that for two functions $f(x)$ and $g(x)$ the limit of the ratio $f(x)/g(x)$ as $x \to a$ is equal to the limit of the ratio of the derivatives $f'(x)/g'(x)$ as $x \to a$. In this case where $i = 1$ (from 10):

$$C_1 = \frac{(\theta_2 - \theta_1)}{2\pi} \tag{14}$$

Having found the coefficients for specific values of θ_1 and θ_2 using equations 13 and 14, the filter series can be calculated by creating a series of NxN arrays. Calculate for each filter element: $e^{j\theta}$ in the first filter multiplied by the harmonic number i, and multiplied by the coefficient C_i. This 'stack' of circular harmonic filters is then added to create the penultimate filter. The final filter is created once the Gaussian function has been reapplied. The final filter can then be used to convolve the relevant corner point of the input image to produce the output for that point. Since θ_1 and θ_2 are likely to be different for each corner point, the final filter needs to be calculated separately for each corner point in the image.

4 Results

4.1 Harmonic Symmetry

When calculating the filter coefficients it becomes clear that the harmonics are symmetrical around harmonic one, rather than around zero as might be expected. This is because of the $(i - 1)$ term in the equation for calculating the coefficients - see equations 13 and 14 in the 3 section. This means that the most accurate results are to be obtained when the harmonic range is chosen to have the median harmonic number as 1, with an equal number of harmonics to either side; from -5 to 7, say. If this constraint is not observed, then the filter values falling within the required range of θ_1 to θ_2 will be skewed and the arguments of the complex filter values will fail to converge to their original values.

4.2 Harmonic Series

There is a compromise to be made when deciding on the number of filters to use in the harmonic series. Since each harmonic is effectively a sampling point taken from a infinite number of possible points, there is an obvious advantage in terms of accuracy in using as many harmonic filters as possible. The disadvantages are in terms of increased processing time and storage space. Figure 8 shows the response in terms of the magnitudes of the moduli of a filter set for a range of θ from $\pi/4$ to $3\pi/4$ for different numbers of harmonics. Note that the Gaussian function has been omitted for the sake of clarity. Figure 8 shows how the response more closely approximates the ideal step function as the number of filters used increases.

4.3 Filter Size

All figures and examples of the filters in this report are shown as 8 x 8 squares, but the filter could be almost any size. This size was chosen because 8 x 8 is the smallest practical size for a working filter. If the filter is too large then there is a risk that output ripples will collide. The mapping between the input layer and the output layer is fixed in a quad-tree architecture, so if the filter is bigger than 2 x 2 then it is possible for the output from two separate input points to overlap, thus confusing the output. Keeping the filter fairly small minimises this possibility.

4.4 Examples

Figures 9 and 10 show the state of the final filters for the three corner points of a simple isosoles triangle like the one in the 'shapes' image - see figure 11(a). Figure 9 shows the filter before the Gaussian function is applied. Figure 10 shows the filter state after the Gaussian has been applied. (a) to (c) show the imaginary parts of the

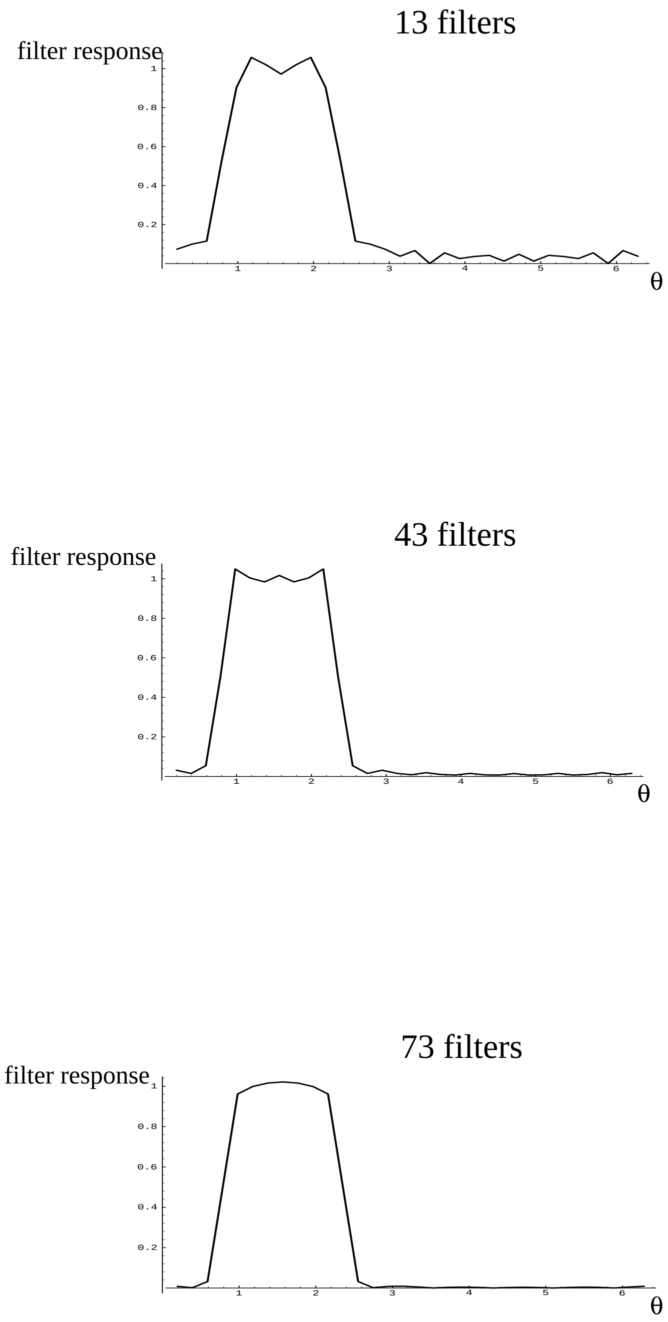

Figure 8: Filter Response to Various Numbers of Harmonic Filters

complex filter values; (d) to (f) show the real parts. The greyscale progresses from black for large negative values, through mid-grey for zero values and on towards white for large positive values. This method of presentation is chosen because it allows the magnitude and direction of the filter values to be appreciated simultaneously. For example: where a pair of spatially corresponding filter values are both the same colour intensity, an angle of $\pi/4$ or some multiple of this is indicated. If the pair are the same colour also, then the angle is either $\pi/4$ (both light) or $-3\pi/4$ (both dark); if the colours are different but of the same intensity then the angle is $3\pi/4$ (real dark, imaginary light) or $-\pi/4$ (real light, imaginary dark). The intensity values give an indication of the magnitudes of the moduluses. This is most easily seen in figure 10 where the Gaussian function has reduced the intensity of all but the most central filter values. It is worth noting that some filter intensities are not quite as expected. This is because the step function is only approximated, as shown by figure 8. For example, the filter values lying on the $\pi/4$ and $3\pi/4$ lines in figure 9(a) and (d) would ideally be more definite than they are.

Figure 11 shows: (a) the clean shapes image, (b) edges generated by the Hopfield network using (a), (c) the shapes image with 6dB of noise added, (d) edges generated by the Hopfield network using (c), (e) the ripple file and (f) edges generated by the Hopfield net using (c) and (e) together. A comparison between (d) and (f) shows that corner and edge detection is enhanced. One missing corner has been recovered from the square, three from the triangle, four from the star, and one from the crescent. Out of 19 corners, 9 were found when the ripple file was not used but 18 were detected when it was used.

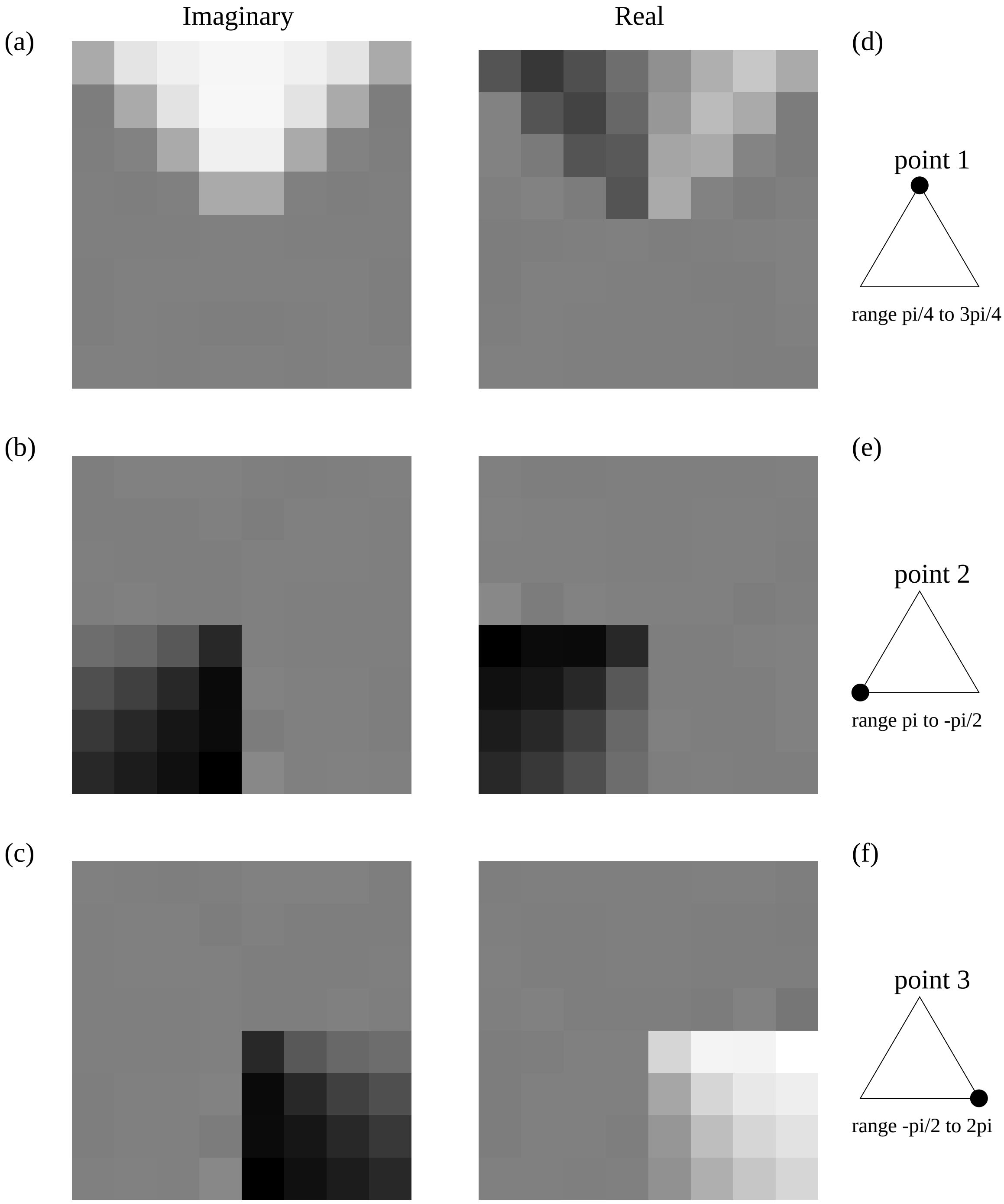

Figure 9: The Ripple Filter for the Triangle Image - No Gaussian

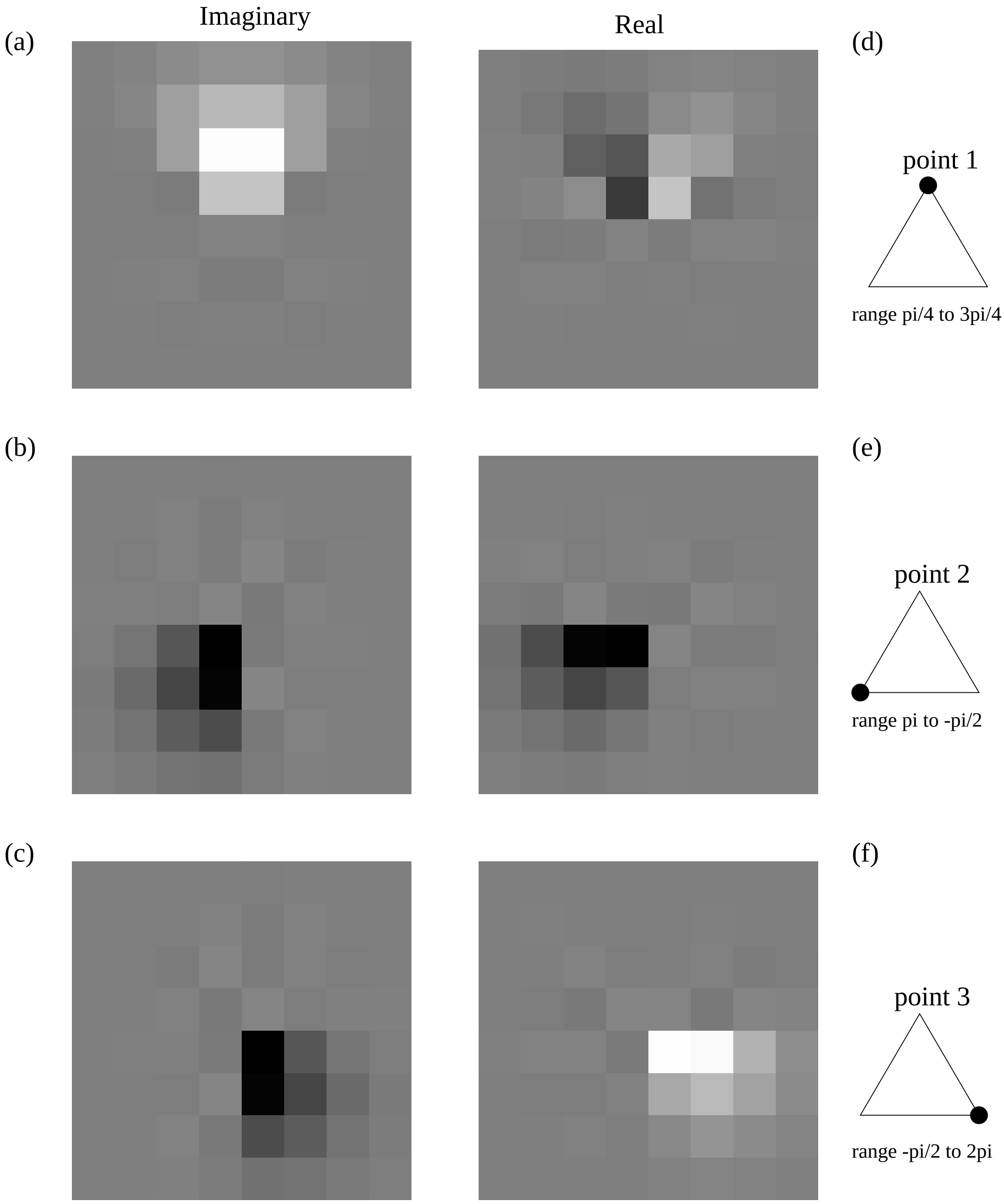

Figure 10: The Ripple Filter for the Triangle Image - Gaussian Included

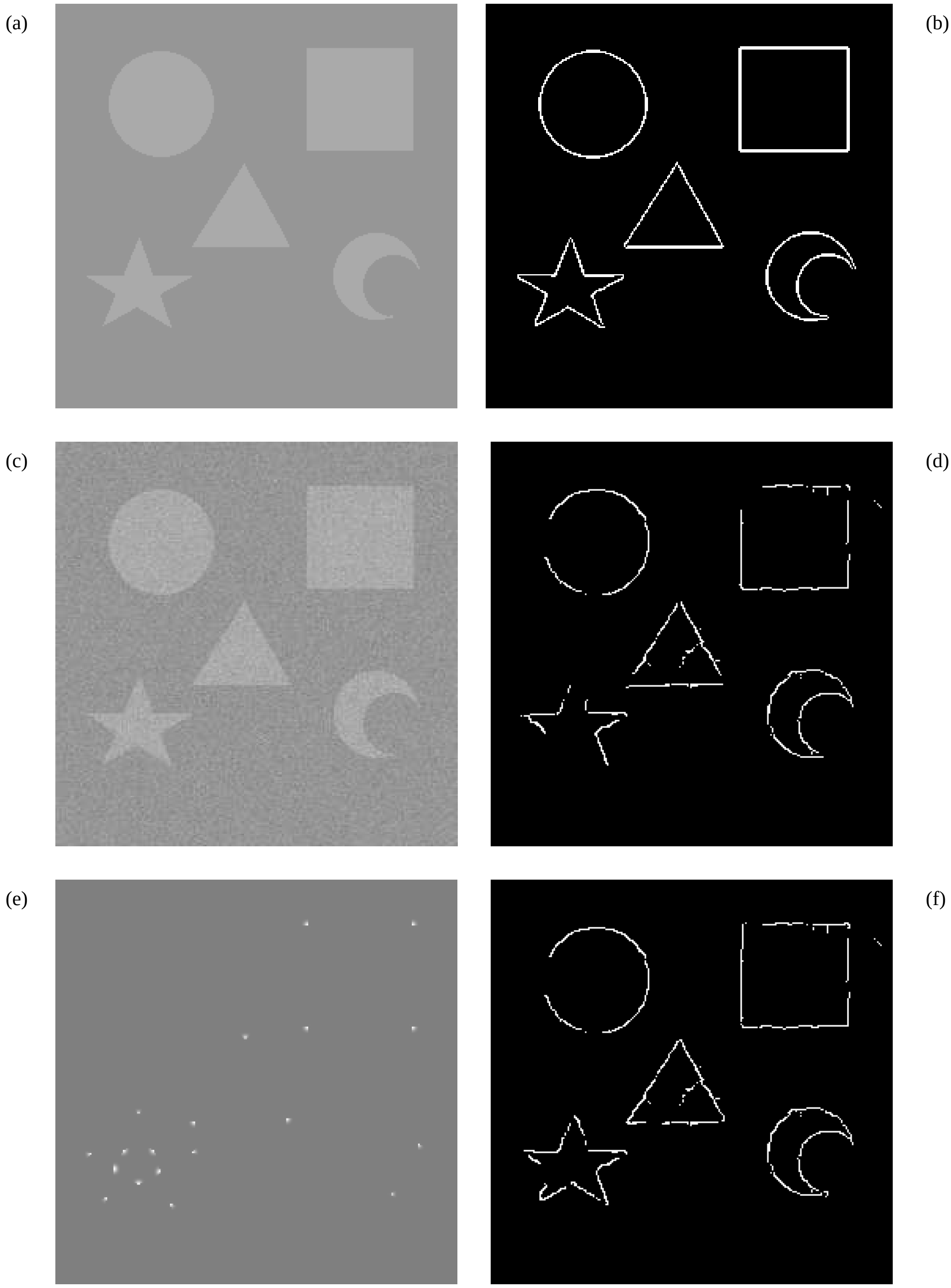

Figure 11: The Shapes Image

5 Discussion

5.1 Circular Harmonic Networks?

5.1.1 Architecture

The question arises as to whether the algorithm for obtaining output of partial ripples could be implemented within a neural network. A literal interpretation would involve allocating a layer of the network for the circular harmonic filters. The harmonic layer could be sectioned into areas, each of which would correspond to a harmonic filter. Each harmonic filter would be connected by NxN connections to the input layer, where NxN is the size of the filter. There would be no connections between the harmonic layers. The final filter would also have connections to each of the the harmonic layers, and the final filter would feed forward into the output layer. If the weights on the input layer to filter lines were set to the harmonic number multiplied by $e^{j\theta}$, and the weights between the final filter and the output layer were set so as to perform the Gaussian function, then the coefficients for the corners could be fed down the input lines as activation. This implies that for every corner point, n coefficients would have to be stored, n being the number of harmonics used. The number n could be small as it would be a fairly simple matter to implement a linear threshold on the output from the final filter so that it would always be either 0 or 1.

5.1.2 Anticipated Problems

So far in this description the problem of implementing a neural net using complex values has been ignored. There are several possible approaches that could be used. Two networks could be implemented side by side; one could perform the real part of the arithmetic, the other handling the imaginary part. Alternatively the two networks could handle the arithmetic in polar form, one performing the arithmetic concerning the moduli, the other handling the arguments. Another problem that would have to be tackled would be that of mapping spatial information from the stored values onto the network. This would need to be implemented at both the input end where the information is stored and at the output end where the information needs to be mapped according to the quadtree architecture.

5.2 Other Networks

It would be possible to achieve the same sort of ripple output using a different type of network altogether. One could take a simple three layer network consisting of an input layer, an output layer and some hidden units. The network could then be trained to give the correct output according to the input pattern which would include the spatial information (ie: where to map the input to) and the ripple information.

The network could then be trained by supervised learning to produce the correct output pattern in the output layer. The main disadvantage of this would be that the training would have to be long and exhaustive. The training set would be very large because of the large number of possible input patterns involved.

5.3 Multiresolution Image Representation

Multiresolution image analysis schemes are currently proving very successful. For this reason it is pertinent to rationalise new ideas for computer vision schemes within a multiresolution context. Corner information could be used to provide an object description at different resolution levels. For example, a leaf might be described at the top level of a multiresolution pyramid as an overall shape, with corners at only the base and tip of the leaf. Further down the pyramid, a description of the corners around a leaf's lobes would become relevant. At the lowest level, corner information pertaining to the serrations around leaf boundaries and details of the veins would be required. The idea of combining corner descriptions with multiresolution image processing does not pose any obvious problems.

5.4 3D Corner Information

It would be possible too extend the output of ripples from 2D to 3D. Another metric would be required to represent the extra axis. The information which would then be required to describe a corner could be stored using a quarternion representation. A quarternion is a four component object which is the sum of a scalar and a vector. Quarternions can be added and subtracted like a four component vectors, and there are mathematical methods for multiplying and dividing them. Quarternions are put to practical use by [8]. Their advantage in the ripple filter context is that they would provide the support for the mathematics involved in the output of ripples as *cones* from concentric spheres of *bubbles* rather than *sectors* of concentric *circles*.

5.5 Dynamic Gaussian Functions

In practice it is sometimes difficult to decide precisely where to position a corner point so that the ripple appears in the output at the correct location. The difficulty arises because there is often a conflict between which part of the filter contains the maximum modulus, and the part of the filter where the arguments are pointing in the correct direction. This is best seen when considering a corner which is bisected by a line orientated at $\pi/2$ or some multiple thereof. The only filter values approaching this value are near the edges of the filter, exactly where the modulus value is weakest. The strongest filter values are in the centre of the filter and have orientations of $\pi/4 + n\pi/2$ - see figure 6 and figure 5. It is therefore not always possible to position the filter so that the ripple is optimally fitted to the corner. For this reason, it would be preferable to have a Gaussian function that could be applied dynamically, rather than having it fixed at the centre of the filter.

5.6 Alternative Corner Functions

There is some doubt as to whether the idea of using partial ripples to represent corners is based on sound principles. The shape of a partial ripple is a sector, and the line that bisects the sector also bisects the corner it maps to. Given that the sector is usually positioned on the corner so that its point is within a node or two of the place where the corner is expected to appear, there is more ripple outside the corner than inside it. This is a problem because what the partial ripple output is 'saying' is that *there is a greater probability that the corner will be found inside this sector than outside it.* This is not true; if the corner is not where it is expected to be, it is equally likely to be found in a place away in any direction. The idea of projecting partial ripples arose because it was found that complete ripples caused problems for the Hopfield net. The Hopfield net would sometimes find false corners when the whole ripple was projected. But this problem was caused by the *orientation* of the superfluous vectors and not by their *presence*. A more satisfactory solution might have been to derive a function to rotate the arguments of the complex filter values so that *all* of them pointed in the correct direction ie: along the line that would bisect the expected corner. This would also avoid the awkwardness involved in positioning the filter for optimal direction and magnitude; the revised filter would always be optimally positioned with its centre over the position of the expected corner. The ripple filter has the advantage of being able to 'guide' the Hopfield network links around a corner. If the filter were revised as just proposed, then the links would be overridden to encourage an abrupt change in direction, more suited to sharp, acute corners than a steep curve. Also, it should be noted that the ripple filter is a fairly loosely constrained method and seems to work well enough in practice, so perhaps it is of little relevance that it cannot be used with great degrees of exactitude. It is a description that can be used to describe any shape, and so gains some merit for its generality and flexibility.

6 Conclusion

The system presented here has worked fairly well and has largely achieved its objective in providing reinforcement for a Hopfield net so that it can detect corners more easily. The mathematical method has proved sound. The features of the mathematical method that have been considered include: the size of the filter, the number of harmonics to use in the harmonic series, and the choice of σ to ensure the correct response for the Gaussian function. The number of harmonics used in the harmonic series has been the single most important factor affecting the accuracy of the results. To achieve a high level of accuracy around 60 filters need to be used. This rather heavy computational load could be reduced by using fewer filters (13 say), and pushing the filter response through a linear threshold, so that the output is always either 0 or 1.

Much of the discussion was concerned with the validity of using sectors of ripples to represent the probable location of the corner. Whilst there are some reservations about this, the method has been shown to be a workable one. Further work with more complicated images could be undertaken, and this would reveal any other potential weak points of the ripple filtering method. It was also shown that there are some problems associated with positioning the filter over the corner. These arise because the Gaussian function is fixed to have its maximum in the centre of the filter. It may be possible to implement a dynamic Gaussian, that allows the 'peak' of the filter to be determined by the requirements of each corner. The disadvantage of this would be that the amount of stored information would be increased.

Refinements and extensions to the work have been discussed. It has been concluded that the ripple filters should be amenable to systems using 3D and multiresolution. Consideration has been given to the possibility of implementing the ripple filters in neural networks. There are anticipated problems for all the methods suggested, but these are not seen as being insurmountable. In particular, the circular harmonic neural network seems promising, provided that the problems foreseen for spatial mapping can be overcome.

References

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

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

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

[4] R. L. Gregory. *The Intelligent Eye.* Weidenfeld and Nicolson, London, 1970.

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

[6] J. E. Hummel and I. Biederman. Dynamic Binding in a Neural Network for Shape Recognition. *Psychological Review*, 99:480–517, 1992.

[7] B. B. Mandelbrot. *The Fractal Geometry of Nature.* Freeman, San Francisco, 1982.

[8] D. B. Tweed and T. Vilis. The Superior Colliculus and Spatiotemporal Translation in the Saccadic System. *Neural Networks*, 3:75–86, 1990.

[9] D. Waltz. Understanding Line Drawings of Scenes with Shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw Hill, New York, 1975.

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