THE UNIVERSITY OF
WARWICK

Original citation:
Wong, J. and Wilson, Roland, 1949- (1990) Eigenvector decomposition of a multiresolution operator. University of Warwick. Department of Computer Science. (Department of Computer Science Research Report). (Unpublished) CS-RR-167

Permanent WRAP url:
http://wrap.warwick.ac.uk/60862

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

A note on versions:
The version presented in WRAP is the published version or, version of record, and may be cited as it appears here.For more information, please contact the WRAP Team at:
publications@warwick.ac.uk

http://wrap.warwick.ac.uk/

Research report 167

EIGENVECTOR DECOMPOSITION OF A MULTIRESOLUTION OPERATOR

JUNE WONG, ROLAND WILSON

(RR167)

Pattern recognition using a multiresolution representation was investigated. This was addressed as an eigenvalue problem. Eigenvector decomposition of a multiresolution operator enabled a low-pass pyramid representation to be expressed in terms of the 'eigenpatterns' of the operator.

The findings show that different image features 'emerge' and can be recognised at different levels of the multiresolution structure, i.e at different resolutions. The level depends on the feature size.

This work has implications in the design of a neural network for pattern recognition, namely that a network could 'learn' the eigenpattern of a multiresolution operator. Patterns recognition processing could proceed in a top-down, hierarchical manner, beginning at a level of coarse features and using information from lower levels in the multiresolution structure to guide the processing of finer detail.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

September 1990

Contents

1 **Learning in Neural Networks** 3
1.1 Multiresolution in Pattern Recognition 5

2 **Simple objects at various resolutions** 7

3 **Eigenvector Analysis of the Low-Pass Transform** 9
3.1 Eigenvalues of the Low-Pass Transform 11
3.2 Oriented Patterns and the Low-Pass transform 13

4 **Conclusions** 16

List of Figures

1 Basic pattern recognition paradigm 20
2 Frequency response of one-dimensional low-pass filter 20
3 Original 'girl' image (256 × 256 pixels) 21
4 Low-pass pyramid representation of the 'girl' image (levels 1 to 6) . . 22
5 Comparing spots with spots: distribution of similarities 23
6 Comparing triangles with triangles: distribution of similarities 23
7 Comparing spots with triangles: distribution of similarities 23
8 Level 1 eigenpatterns 24
9 Level 1 eigenpatterns weighted by their eigenvalues 25
10 Level 3 eigenpatterns weighted by their eigenvalues 26
11 Level 6 eigenpatterns weighted by their eigenvalues 27
12 Oriented Gaussian patterns (L to R: original 'blobs'; Levels 1 - 6 of low-pass pyramid) 28
13 Magnitudes of first three eigenvalues 29
14 Amount of energy retained in first few eigenvectors 30
15 Magnitudes of λ_0 for N = 8, 16, 32 31
16 Preservation of anisotropy for Gaussian ellipse and curve 32
17 Preservation of anisotropy for Gaussian ellipses at various orientations 33
18 Errors in various Gaussian blobs for N = 8 34
19 Errors in circular Gaussian blobs for N = 8 and N = 4 35

Introduction

Vision seems simple to us, the users. We are far better at perceiving objects in natural scenes than modern computers, yet we are certainly not quicker or more precise.

People outperform today's computers because the brain employs a basic computational architecture that is more suited to dealing with the natural information processing tasks that we are so good at. These tasks generally require the simultaneous consideration of many pieces of information or constraints. Each constraint may be imperfectly specified or ambiguous, yet each can play a decisive role in determining the outcome of processing.

Artificial neural networks have received considerable interest in recent years since they are able to tolerate ambiguities—as is the human brain. Each 'neuron', in such systems, is a very simple computational element, yet a densely interconnected network of these elements can achieve good performance in areas such as speech and image recognition.

The importance of multiresolution techniques to several image processing applications is discussed in Section 1, after a brief description of some of the principles behind neural networks. At present, most neural network research into image analysis takes no account of the need for multiresolution methods. It is intended that this work should examine the potential application of a multiresolution approach to pattern recognition problems, with a view to implementing a neural network solution.

The second section deals with experimental findings arising from the use of a multiresolution low-pass pyramid data structure to represent simple shapes.

The search for patterns which are invariant to the multiresolution operator is tackled from an eigenvector decomposition point of view in Section 3. Experimental results are presented to illustrate the points made concerning the way pattern detail can be represented within a pyramid of low-pass filtered images.

The concluding remarks in Section 4 comment on the way a multiresolution approach to pattern recognition could proceed.

1 Learning in Neural Networks

Artificial neural net models are composed of densely interconnected simple computational elements. In this respect their structure is based on present understanding of biological nervous systems. The computational elements, or nodes, are interconnected via weights that are typically adapted during use to improve performance based on current results. The simplest example of a node is one which sums N weighted inputs and passes the result through a nonlinearity.

Generally speaking, the behaviour of a network of artificial neurons is as follows:

- Each node or unit, u_j, has an associated activation value $a_j(t)$.

- An output function (often some sort of threshold function) maps the current state of activation to an output signal, $o_j(t)$
  That is,
  $$o_j(t) = f_j(a_j(t))$$

- A set of such units feed their output signals, via weighted connections, to a unit in the next layer. This unit, u_i, sums the weighted signals coming in to it and combines them with its current state of activation to produce a new state of activation. In vector notation, the new state for the entire network is given by:
  $$\mathbf{a}(t+1) = \mathbf{F}(\mathbf{a}(t), W\mathbf{o}(t))$$

- The matrix W represents the pattern of connectivity; the absolute value of the entry w_{ij} represents the strength of the connection from u_j to u_i.

Neural net models have greatest potential in applications involving many competing hypotheses which are pursued in parallel; one such application is visual pattern recognition.

Adaptation, or learning, is a major focus in neural net research. The ability to adapt and continue learning is essential in pattern recognition, where training data are limited and new images are constantly encountered. Learning, in this context, requires the modification of the pattern of connectivity as a function of experience. This entails adapting the strengths of the connections between units.

Neural net adaptation algorithms fall into two camps: supervised and unsupervised learning.

1. Supervised Learning
   The network is provided with side information or labels (by a 'teacher'). Nets trained with supervision act as either pattern classifiers or associative memories, and are typically used with incomplete input patterns, or those corrupted by noise or some other process.

2. Unsupervised Learning
   No information concerning the correct class of the input pattern is provided during training. Nets trained without supervision can be used to vector quantise or cluster the inputs. They are capable of compressing the amount of data without losing important information. The net result is that individual units 'learn' to 'specialise' on specific patterns and thus become 'feature detectors'.

Lippman [8] has written a very comprehensible review article in which he describes six neural net models that can be used for pattern classification. He gives examples of the use of both supervised and unsupervised training. See also the review by

Carpenter [4], who uses adaptive filter notation to describe some basic neural network modules.

The work of neuropsychologist Donald Hebb, in 1949, has spawned many neural network training rules - see, for example, [11]. Based on Hebb's ideas on memory and learning, these rules all employ the principle that the change in connection strength contains a term proportional to the product of input and output activities at that connection, ie

$$\Delta w_{ij} = \eta a_i o_j$$

where a_i is the activation of u_i and o_j is the output signal from u_i.

Hebbian learning rules modify connection strengths according to the degree of correlated activity between the input and output units [7]; they tend to maximise the variance of the output units [12] [4].

Of particular note in the application of Hebbian-type rules in unsupervised learning has been the work of Linsker [7], in 1988, and Sanger [12], in 1989.

Linsker used a self-organising neural net to model how a perceptual system could develop to recognise specific features of its environment, without being told which features it should analyse. He used a Hebbian algorithm to adapt the connection strengths and achieved minimum mean squared error in the linear reconstruction of the input patterns, given the activities of the output nodes.

Sanger employed the same basic Hebbian learning algorithm, based on one first proposed by Oja in 1982. He proved that his 'Generalised Hebbian Algorithm' computes the eigenvectors of the input autocorrelation matrix. Sanger discussed the usefulness of eigenvector decomposition applied to neural networks, and demonstrated the power of networks trained according to Hebbian principles when applied to image coding, feature extraction and texture analysis.

1.1 Multiresolution in Pattern Recognition

The basic paradigm of a visual pattern recognition system is shown in Figure 1. In general, the goal of such a system is to analyse images of a given scene and to recognise the content of the scene [19]. The aim is to obtain a representation consisting of a set of symbols with appropriate relationships, eg "the object A is of class C and is located at position X". It is widely recognised that this fundamental problem of image analysis is affected by the Uncertainty Principle [18] [14]—that is, the ability to resolve features spatially is constrained by the extent to which they may be resolved spectrally.

The derivation of a symbolic description of an image necessitates the extraction of meaningful features from the image data, such as lines, edges and texture elements. These features, which must capture the important characteristics of the input data, form the input to subsequent processing.

Both objects and features occur at many spatial scales. Marr [9] describes the 'hierarchical organisation' of an image:

> "The spatial organisation of a surface's reflectance function is often generated by a number of different processes, each operating at a different scale."

Marr explains the task of vision using oriented lines and edges and blobs (ie local features) as a basis. He points out that there is considerable psychophysical and neurophysiological evidence to suggest the importance of oriented line and edge features in low-level mammalian visual processing [6].

The first step in detecting lines and edges is to blur the image. This is achieved by performing a low-pass filtering operation on the image, thereby imposing an upper limit on the rate at which intensity changes can take place. Since these changes occur at different scales [9] in the original image, their optimal detection requires operators of different sizes, ie filters of different bandwidths.

This multiresolution approach enables the spatial consistency of the outputs from several different scales of filters to produce a more reliable estimate of feature position than is possible from a single filtering operation.

A pyramidal data structure is used to represent the image data over a range of spatial resolutions. This type of structure has found increasing usage in image processing since it can match the multiresolution representation of images—a representation recognised as valuable in a number of applications [2] [5] [16].

In the present work, a 'low-pass pyramid' is generated by filtering the image with low-pass filters having a two-dimensional 'raised-cosine' frequency response, and then subsampling. The frequency response of the filter is linearly separable; its one-dimensional form is shown in Fig. 2.

The base level of the pyramid is the original image; each higher level uses a filter with half the bandwidth of the preceding level. This obtains versions of an image at successively lower spatial resolutions (but higher spatial frequency resolutions), which require fewer spatial samples and are thus 'smaller'.

The image is therefore represented as a series of low-passed images, each sampled at successively lower densities. The result is a self-similar structure which exhibits localisation in both space and spatial frequency. Each level of the low-pass pyramid contains a blurred version of the original image; the degree of blurring, which varies according to the pyramid level, is governed by the bandwidth of the low-pass filter. Fig. 3 shows the original image of the girl, and levels 1 to 6 of its pyramidal representation are given in Fig. 4.

On the whole, neural networks performing visual pattern recognition tasks have an arbitrary scale imposed upon them. However, the many examples of the importance of scale in image processing and analysis [2] [3] [5] [16] [15] suggest using a multiresolution pyramid as input to a neural net to enable pattern recognition in the manner of 'successive refinement'.

Schalkoff comments: [13]

> "A hierarchical processing approach parallels the operation of the human visual system in that processing proceeds from a coarse descriptive level to levels of increasing refinement. The hierarchical approach is applicable to the tasks of segmentation, feature extraction, description and matching."

One question of interest concerns the effects on patterns and features of this multiresolution operator. Intuitively, one would expect that a spatial scale exists at which individual objects 'merge' and, therefore, at some levels of the pyramid single objects are indistinguishable.

Some elementary investigations into the behaviour of images of several objects in the low-pass pyramid representation are described in the following section.

2 Simple objects at various resolutions

In order to investigate the properties of the low-pass pyramid representation, images were synthesized by placing a simple object, or pattern, at a number of spatial positions within a 512 × 512 pixel area. Four classes of simple object were used: spots, squares, bars and triangles, of similar energy.

For each 512 × 512 pixel image a low-pass pyramid of 7 levels was constructed: level 0 (the original image) has greatest spatial resolution; level 6 (8 × 8 pixels) has lowest spatial resolution.

Each synthetic image of 512 × 512 pixels can be considered to be made up of an 8 × 8 array of 'cells', each of size 64 × 64 pixels. Thus, each pixel at the top level of the pyramid (level 6) corresponds to one cell at the bottom level (level 0).

Each cell contains one simple object. The position of the object in relation to its cell boundaries was varied in a systematic way over the entire image. The 'exemplar' for each class of object was the cell with the object centred within it.

The low-pass pyramid representations were generated in order to address some fundamental questions:

- do all objects in a given image 'appear' at the same pyramid level?
- do all the shapes 'appear' at the same level?
- is the shape of the object recognizable at this level?

A quantitative measure of the similarity, or 'overlap' between two n × n regions of pixels is given by the inner product of their intensity values, say, $f(x,y)$ and $g(x,y)$, where $0 \leq x, y < n$.

That is

$$\begin{aligned} R_{fg} &= < f(x,y), g(x,y) > \\ &= \sum_{(x,y)} f(x,y)g(x,y) \end{aligned} \tag{1}$$

where R_{fg} denotes the similarity measure, or correlation.

The results which follow were obtained by computing the inner product between the exemplar and spatially shifted objects (for each class). In other words, we investigated how well objects at various spatial locations in a blurred image matched a 'prototype' filtered image.

This was repeated to examine the similarities between blurred objects from different classes, and to determine whether the magnitude of the inner product could be used as a criterion for differentiating between such objects, given only their blurred representation in the low-pass pyramid.

Results

At level 4 the image size was 32 × 32 pixels and each cell occupied 4 × 4 pixels; this was the lowest spatial resolution at which the objects were visible individually. However, at this level the spots and squares were identical; the triangles and bars were seen as 'blobs', slightly different in each cell. It was not possible to identify their original shape , although there were differences between the triangles and the bars.

At the next higher spatial resolution (ie image size of 64 × 64 pixels) all of the shapes were recognizable.

The R values (see Equation 1) for the spots and triangles are presented, since these two objects can be considered to have the least in common of the classes used here. The distribution of the R values is given in histogram form for the following comparisons:
spots with spots—Fig. 5;
triangles with triangles—Fig. 6;
spots with triangles—Fig. 7.

The results shown were all computed using the representations from level 4 of the pyramid; ie the objects were visible but their shape was indeterminate due to the blurring by the low-pass filter.

It is obvious from these histograms that a considerable spread of R values was obtained, even when objects from the same class were compared. Hence, the magnitude of the inner product did not convey sufficient information to differentiate objects from different classes. It follows that a neural unit trained to identify one such pattern would be unable to discriminate this pattern from the other patterns.

This raises the next question—whether any patterns exist (and if so, what are

they?) that remain unchanged under low-pass filtering and therefore could be recognised at any spatial resolution.

3 Eigenvector Analysis of the Low-Pass Transform

The question raised in the preceding section concerned patterns or objects which are invariant to low-pass filtering. Invariance in this context means that a pattern at level n of the low-pass pyramid can be projected down to level 0 (maximum spatial resolution) by suitable interpolation, and then taken back up to level n again without being changed 'significantly'.

Finding such patterns is an eigenvector problem which entails representing the filtering operation as a linear transform. In view of the fact that the 2-d low-pass filter is linearly separable, it is possible to solve the eigenvalue/eigenvector problem in the one-dimensional case; the two-dimensional solutions are then given by the products of the corresponding one-dimensional solutions.

Notation

Matrix notation is adopted as being the most concise. The following conventions have been adopted:

1. Linear operators are represented by capital letters and are taken to be $2^m \times 2^m$ matrices
$$A = (a)_{kl}$$
$$0 \leq k, l < 2^m$$

2. Signal vectors are $2^m \times 1$ column vectors and are denoted by lower case boldface letters, eg $\mathbf{v}$.

3. Common matrix operations are:

   $'$: transpose
   $*$: adjoint (conjugate transpose)

   Thus, the inner product of two vectors will be written as:
$$\mathbf{v}^*\mathbf{u} = \sum_{k=0}^{2^m-1} u_k v_k^*$$

The matrix F is the discrete Fourier transform (DFT) operator; F^* is the inverse DFT operator.

T_N is a truncation operator; it ensures spatial limiting and is defined by:

$$\begin{aligned} T_{Nkl} &= \delta_{kl} \quad 0 \leq k < N \\ &\qquad\quad 2^m - N \leq k < 2^m \\ &= 0 \quad \textit{otherwise} \end{aligned}$$

Note also that $T'_N = T_N$.

The low-pass pyramid transform, in one-dimensional space, can be modelled:

$$T'_N F^* H^2_{LP} F T_N \mathbf{v} = \lambda \mathbf{v} \tag{2}$$

Since T_N (the truncation operator) is also a projector [10] then

$$T'_N T_N = T_N T'_N = T_N$$

Premultiplying both sides of Equation 2 by T_N obtains:

$$\begin{aligned} T_N F^* H^2_{LP} F T_N \mathbf{v} &= \lambda T_N \mathbf{v} \\ \lambda \mathbf{v} &= \lambda T_N \mathbf{v} \end{aligned}$$

The non-trivial solution is

$$\mathbf{v} = T_N \mathbf{v} \tag{3}$$

This implies that

$$v_k = 0 \qquad N \leq k < 2^m - N \tag{4}$$

H_{LP} is the frequency response operator of the low-pass filter. Since it is used twice (as a low-pass and as an interpolation filter) it appears as H^2_{LP} in Equation 2.
If we write

$$F^* H^2_{LP} F = H$$

then

$$(H)_{kl} = h_2(k - l) \tag{5}$$

where

- $h_2(n), \quad 0 \leq n < 2^m$ are the filter impulse response coefficients
- $(k - l)$ is calculated modulo-2^m.

Thus Equation 2 becomes

$$THT\mathbf{v} = \lambda \mathbf{v} \tag{6}$$

It can be shown that

$$(THT)_{kl} = (H)_{kl} = h_2(k - l) \tag{7}$$

$$0 \leq k, l < N, \quad 2^m - N \leq k, l < 2^m$$

In other words, the operator represents the effect of taking a spatially localised or truncated pattern (of 2N samples) at some level of the pyramid, projecting it onto the image and then constructing the pyramid based on that image. Eigenvectors of these operators are 'eigenpatterns' of the multiresolution representation—only their length changes when they are subjected to these operations. The eigenpatterns of the pyramid are, in a sense, invariants of the representation.

Now, the H operator is a $2^m \times 2^m$ matrix. If all the zero-valued rows and columns are removed we are left with a $2N \times 2N$ matrix; correspondingly, the eigenpattern v can be reduced to $2N$ dimensions without any loss of information (see Equation 4).

In fact, these eigenpatterns are closely related to a set of vectors well known in signal processing and used in a number of image processing applications—the prolate spheroidal sequences. The finite prolate spheroidal sequences (FPSS) [17] combine spatial and frequency domain locality in an optimal way, since they maximise the energy in a finite interval of one domain while truncating in the other. The FPSS is the eigenvector, $\mathbf{g}_0$, in the following problem:

$$BT\mathbf{g}_0 = \lambda_0 \mathbf{g}_0 \tag{8}$$

where T is the truncation operator and B is the bandlimiting operator, defined as

$$B = F^* T F \tag{9}$$

and λ_0 is the largest eigenvalue. In this case, the FPSS is defined to be exactly bandlimited and optimally concentrated in a finite spatial region.

The problem addressed in Equation 2 differs in that bandlimiting is achieved by using the low-pass filter, H_{LP}, with a 'raised-cosine' frequency response, whereas the FPSS problem requires an 'ideal' rectangular filter. Although it is possible to use ideal low-pass filters to construct a pyramid, these have large sidelobes in the spatial domain which make them insufficiently localised for most applications.

3.1 Eigenvalues of the Low-Pass Transform

The reduced operator THT is a real symmetric matrix; hence, the 2N eigenvalues and eigenvectors are purely real, and the eigenvectors form a mutually orthogonal basis [10].

The eigenvectors are the one-dimensional patterns which undergo only multiplication by a scalar under the low-pass transform. The amount of energy retained by each eigenvector after filtering is given by the magnitude of its corresponding eigenvalue, in the sense that

$$\mathbf{v}^* A \mathbf{v} = \lambda \mathbf{v}^* \mathbf{v} = \lambda$$

The graph plotted in Fig. 13 shows the magnitudes of the first three eigenvalues $(\lambda_0, \lambda_1, \lambda_2)$ against the bandwidth of the low-pass filter operator, H_{LP}.

These results were obtained using $2^m = 256$ and N = 8; they show that, for a given filter bandwidth, the magnitudes of the eigenvalues, $| \lambda_k |$, decrease very quickly with increasing index, k. Therefore, most of the energy is concentrated in the first few eigenvectors; indeed, at higher levels in the pyramid (ie narrower bandwidths) only the first eigenvalue, λ_0, is 'significant'.

The transform as a whole passes a certain amount of the total signal energy. At a particular pyramid level, this amount is given by the sum of all 2N eigenvalues. A percentage of this 'transformed energy' is passed by each eigenvalue, according to its magnitude. Previous findings (see Fig. 13) have shown that the number of eigenvalues having 'significant' magnitude is dependent on the pyramid level. Therefore, most of the 'transformed energy' is passed by only a proportion of the total number of eigenvalues; this is shown in Fig. 14. As an extreme example, consider the case at level 6: only about 6% of the eigenvalues account for over 99% of the 'transformed energy'. As might be expected, increasing the bandwidth requires a larger proportion of the number of eigenvalues in order to maintain the same percentage of 'transformed energy'.

Fig. 15 shows the magnitude of λ_0 as a function of bandwidth for N = 8, N = 16 and N = 32. These plots show that each doubling of N has the same effect as doubling the bandwidth. Therefore, for a particular eigenvalue, the product of N and bandwidth is a constant.

This implies that there is more energy at higher pyramid levels if N is increased. The value of N indicates the initial level from which the pattern is projected onto the image in order to construct the pyramid. Thus, a larger value of N is equivalent to 'starting off' from a lower level of the pyramid structure.

The Two-Dimensional Case

Because the 2-d operator is separable, its eigenvalues and eigenvectors are easily related to the 1-d case.

The two-dimensional eigenvalues are given by

$$\lambda_{ij} = \lambda_i \lambda_j$$

If, in the one-dimensional case, the 2N eigenvectors of the transform are given by

$$\mathbf{v}_i, \quad 0 \leq i < 2N$$

then the $2N \times 2N$ two-dimensional eigenvectors can be written as

$$e_{ij}(x, y) = v_i(x) v_j(y)$$

These form a mutually orthogonal basis because any 2-d 'pattern' can be expressed as a linear combination of these 2-d basis vectors. For a given bandwidth, if the 2-d eigenvectors are weighted by the corresponding 2-d eigenvalues, the result is equivalent to passing the 'patterns' through a low-pass filter of that bandwidth. Only some of the patterns 'survive', since the eigenvalue distribution favours the first few eigenvectors.

The 2-d eigenvectors when N = 8 are shown in Figs. 8 – 11. The first of these figures shows all of the eigenvectors (or eigenpatterns) at level 1 of the pyramid. The remaining figures show how the number of significant eigenpatterns varies with pyramid level when they are weighted by their corresponding eigenvalues; they also give an indication of how many significant eigenpatterns there are at different pyramid levels.

Thus at the lowest bandwidth, ie the highest pyramid level, only one significant eigenpattern is seen (Fig. 11). As the bandwidth increases on successive levels, more patterns having significant residual energy can be seen. As might be expected, the effective dimensionality of the signal increases—more and more details become visible (see Fig. 9). This is a direct consequence of the distribution of eigenvalues at different bandwidths, as discussed in Section 3.1, since a greater proportion of the eigenvalues are large enough to be 'significant' as the bandwidth increases. For example, at high pyramid levels only the first eigenvalue is significant. Consequently the low-pass filtering at this level 'destroys' those eigenpatterns which convey detail.

3.2 Oriented Patterns and the Low-Pass transform

In order to investigate the effects of the low-pass pyramid transform on oriented patterns and objects, a set of synthetic images was designed, based on elliptical, curved elliptical, and circular Gaussian intensity distributions. Each pattern occupies a $2N \times 2N$ pixel area; the patterns are shown in the first column of Fig. 12.

Any pattern, $f(x, y)$, can be expressed as a weighted sum of the orthonormal basis:

$$f(x,y) = \sum_{(i,j)} \alpha_{ij} e_{ij}(x,y)$$

with

- α_{ij} : coefficients in the linear expansion;
- $e_{ij}(\cdot)$: 2-d eigenvector (orthonormal basis).

The weights, α_{ij}, are given by the inner product of the pattern and the 2-d eigenvector; that is:

$$\begin{aligned} \alpha_{ij} &= < f(x,y), e_{ij}(x,y) > \\ &= \textstyle\sum_{(x',y')} f(x',y') e_{ij}(x',y') \end{aligned}$$

Now, if the eigenvectors, $e_{ij}(x,y)$, are weighted by their 2-d eigenvalues, λ_{ij}, the resulting pattern is given as

$$\hat{f}(x,y) = \sum_{(i,j)} \alpha_{ij}\lambda_{ij}e_{ij}(x,y)$$

or, equivalently

$$\hat{f} = (THT)f$$

and shows how much of the shape and orientation of the original pattern, $f(x,y)$, is preserved under the low-pass transform.

With reference to Fig. 12, the representations at pyramid levels 1 to 6 are shown in columns 1 to 6, with the original Gaussian patterns in column 0.

As might be expected, at levels 3 and above all of the patterns are seen as roundish 'blobs'—the low-pass operator has destroyed the 'orientedness' of the originals.

It has already been shown (Fig. 11) that eigenpatterns with small eigenvalues lose most energy under the low-pass transform. This is the case for all but the first few eigenpatterns, particularly at higher levels (ie lower bandwidths) of the pyramid. Obviously, the eigenpatterns necessary to convey the 'orientedness' of an object are lost after the first couple of levels, whereas enough eigenpatterns survive to indicate the 'presence' of an object at much higher levels.

The next section deals with a more quantitative analysis of the pyramid representation of oriented objects, in terms of the errors introduced and the amount of orientedness lost due to the low-pass operator.

Preservation of Anisotropy

The theory of moments of inertia [1] permits analysis of a system, often encountered in mechanics, of point masses rotating about a fixed axis. It is possible to determine the principal axes of such a system, and to obtain an estimate of the degree of mass concentration. A related problem is an analysis of the dispersion of a set of (weighted) points with respect to the centroid of the set, and the determination of the principal axes [3].

Thus, by viewing the Gaussian pattern as a set of weighted points it is possible to use the moments of inertia to determine the principal axes of the filtered pattern and hence obtain a quantitative measure of anisotropy , or 'orientedness'.

Define the set of points by the position vector $[xy]'$; the origin is assumed to be the centre of the 2N × 2N pixel area in which the Gaussian pattern, $f(x,y)$, is located.

The 'inertia tensor' is then given by [1]

$$T = \sum_{(x,y)} \mid f(x,y) \mid [xy]'[xy]$$

where $| f(x,y) |$ is the 'weight' associated with the point whose position vector is $[xy]'$.

Since T is real symmetric, its eigenvectors are orthogonal, and it can be shown that the principal axis is given by the eigenvector corresponding to the smallest eigenvalue, λ_0; it defines the direction of minimum dispersion. In other words, the principal axis is given by the eigenvector corresponding to the largest eigenvalue, λ_1.

A measure of anisotropy, δ, of the pattern can be obtained as follows:

$$\delta = \frac{(\lambda_1 - \lambda_0)}{(\lambda_1 + \lambda_0)}$$

By comparing the anisotropy measures before and after the pyramid operator, it is then possible to see how much of the initial anisotropy remains after the operator. The ratio

$$\rho = \frac{\delta'}{\delta} \tag{10}$$

where

- δ' : anisotropy of the filtered pattern
- δ : anisotropy of the original pattern.

is a convenient measure of how much anisotropy of the Gaussian patterns is preserved at various levels in the low-pass pyramid.

Similarly the overall error introduced by the pyramid operator can be defined to be the sum of the squared differences between the original Gaussian pattern and the low-pass filtered version, over all the pixels in the 16 × 16 area containing the pattern. That is, the error, E is given by:

$$\begin{aligned} E &= \sum_{(i,j)} (\alpha_{ij} - \alpha_{ij}.\lambda_{ij})^2 \\ &= \sum_{(i,j)} \alpha_{ij}^2 (1 - \lambda_{ij})^2 \\ E &= \| f - \hat{f} \|^2 \end{aligned} \tag{11}$$

Results

The amount of 'orientedness' retained (cf Equation 10) at various levels of the pyramid is shown in Fig. 16 for the Gaussian ellipse and curve (the orientation of a circular object is zero both before and after filtering.). There is almost no orientation left by level 3, and none at all at higher levels of the pyramid. This confirms the qualitative assessment of Fig. 12.

Gaussian ellipses at three different orientations are compared in Fig. 17. The slight difference between these is due to the cartesian separable nature of the pyramid operator; if a more nearly circular filter had been used, this would not have been seen.

As might be expected, thinner versions of the Gaussian patterns at the same orientations gave rise to identical results.

The amount of error introduced (cf Equation 11) by the low-pass operator is shown in Fig. 18 for circular, elliptical and curved Gaussian 'blobs'. Unlike the anisotropy measurements, the orientation of the elliptical blob had no effect on the errors introduced. Unsurprisingly, the thinner blobs suffered greater errors at all pyramid levels in comparison to their 'large' counterparts.

All of the above results relate to a spatial dimension of N = 8, ie there were 16 eigenvectors of 16 elements. It is of interest to note the effect of changing the dimensionality of the pattern: Fig. 19 compares the errors introduced into the low-pass representations of circular blobs for N = 8 and N = 4. (In the latter case the circular object was scaled to fit within an 8 × 8 pixel area.) Thus, twice the bandwidth is required to keep the error constant if the value of N is halved.

The most significant finding is that there is a simple relation between the size of an isotropic feature and the pyramid level at which its anisotropy emerges (the 'orientation level') and this is not the same as the level at which the feature can be seen at all (the 'detection level'). This may be expressed quantitatively as follows:

$$\begin{aligned} \text{Detection level} &\approx \log_2 N \\ \text{Orientation level} &\approx \text{detection level} + 1 \end{aligned}$$

4 Conclusions

In mammals, visual information is processed in stages. Edges are detected and edge orientation is analysed in the earlier stages; later processing is concerned with more complex aspects of form. The intention of this work has been to investigate how progressively more and more detail can be represented, such that pattern recognition processing could begin at a level of very coarse features and proceed, making use of current information to guide the processing of finer detail.

With this in mind, a multiresolution linear operator was examined to find out what sort of patterns can and cannot be represented at various levels of a low-pass pyramid.

There is a pyramid level at which any simple image pattern—circle, line or curve—will first appear. At this level, all such patterns appear as more or less circular blobs. The level depends only on the feature size, as one would expect. The relation between size and pyramid level is that, roughly speaking, halving the size of a feature requires going down one level in the pyramid in order for the feature to first appear.

At levels below this, features originally having some anisotropy are preserved, so that orientation as a feature property can be defined and measured in a consistent way. Still further down, new properties, such as curvature emerge and so a broader class of patterns can be meaningfully represented.

It follows that in designing neural networks for multiresolution pattern recognition, it is necessary to allow patterns to 'emerge' or be recognised in a top-down hierarchical manner, in which first the existence of 'something' (ie a blob) is detected. This can be refined by adding patterns from lower levels, but now conditioned on those features already detected. This is similar in many ways to the segmentation by Spann and Wilson [16].

Proposals for future work

A network of artificial neurons could 'learn' the invariants of a multiresolution operator. It is not yet apparent whether a supervised or unsupervised learning algorithm would be most appropriate in this context. Linsker [7] and Sanger [12] both used unsupervised learning and succeeded in training networks to compute the eigenvectors of the correlation matrix of the inputs to a layer of neurons. In this case, however, the eigenpatterns of a multiresolution operator are sought, together with ways of using them to successively refine the pattern recognition process. The design and development of such a network and its learning regimes offers scope for a great deal of interesting work to follow.

References

[1] A. I. Borisenko and I. E. Tarapov. *Vector and Tensor Analysis with Applications.* Dover Publications, New York, 1968.

[2] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Comp.*, COM-31:532–540, 1983.

[3] A. Calway. *The Multiresolution Fourier Transform: A general Purpose Tool for Image Analysis.* PhD thesis, Department of Computer Science, The University of Warwick, UK, September 1989.

[4] G. A. Carpenter. Neural network models for pattern recognition and associative memory. *Neural Networks*, 2:243–257, 1989.

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

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

[7] R. Linsker. Self-organization in a perceptual network. *Computer*, 21:105–117, 1988.

[8] R. P. Lippman. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22, April 1987.

[9] D. Marr. *Vision.* Freeman Press, San Fransisco, C.A., 1982.

[10] E. D. Nering. *Linear Algebra and Matrix Theory.* John Wiley and Sons, New York, 1970.

[11] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing*, volume 1:Foundations. MIT Press, Cambridge, Massachusetts, 1986.

[12] T. D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, (2):459–473, 1989.

[13] R. J. Schalkoff. *Digital Image Processing and Computer Vision.* John Wiley and Sons, New York, 1989.

[14] D. Slepian. On bandwidth. *Proceedings of the IEEE*, 64(3):292–300, March 1976.

[15] M. Spann. *Texture Description and Segmentation in Image Processing.* PhD thesis, Department of Electrical Engineering, The University of Aston in Birmingham, UK, September 1985.

[16] M. Spann and R. G. Wilson. A quad-tree approach to image segmentation which combines statistical and spatial information. *Pattern Recognition*, 18(3/4):257–269, 1985.

[17] R. Wilson. Finite prolate spheroidal sequences and their applications I,II. *IEEE Trans. P.A.M.I.*, PAMI-10, 1988.

[18] R. G. Wilson and M. Spann. *The Uncertainty Principle in Image Processing.* Pattern Recognition and Image Processing Series. Research Studies Press Ltd, 1988.

[19] T. Y. Young and K. S. Fu, editors. *Handbook of Pattern Recognition and Image Processing.* Academic Press, San Diego, California, 1986.

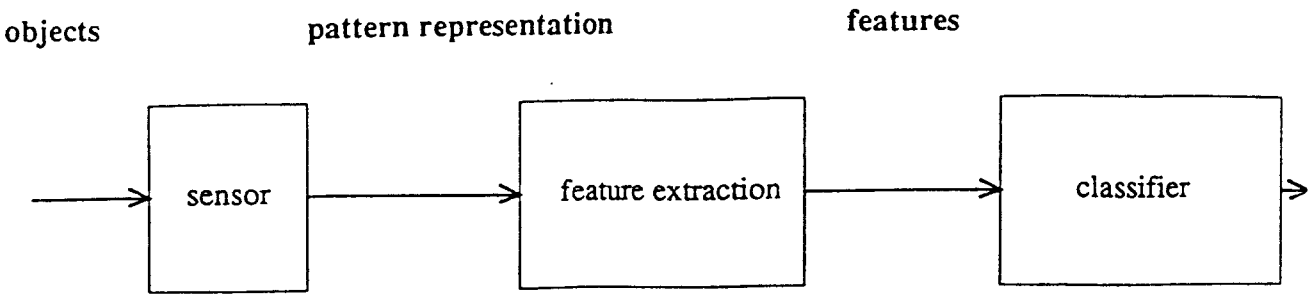

Figure 1: Basic pattern recognition paradigm

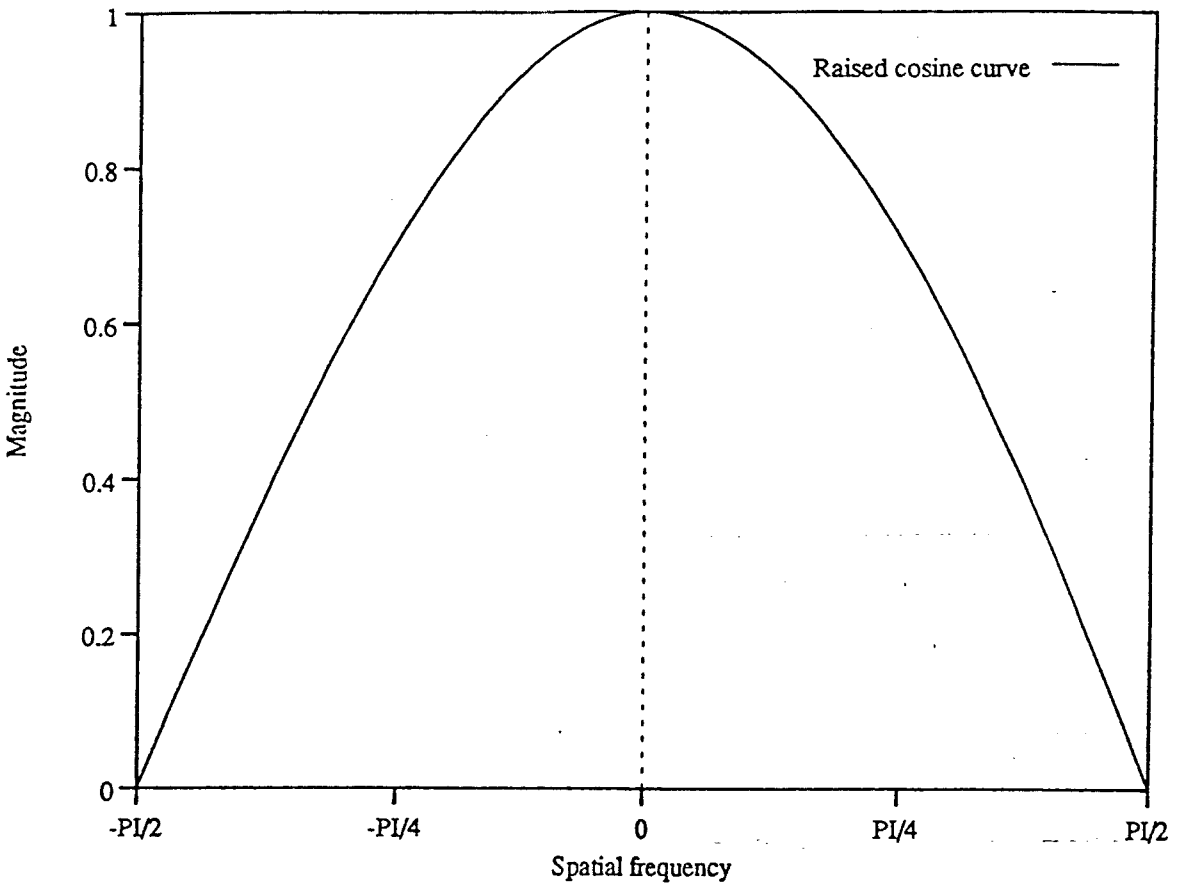

Figure 2: Frequency response of one-dimensional low-pass filter

Figure 3: Original 'girl' image (256 × 256 pixels)

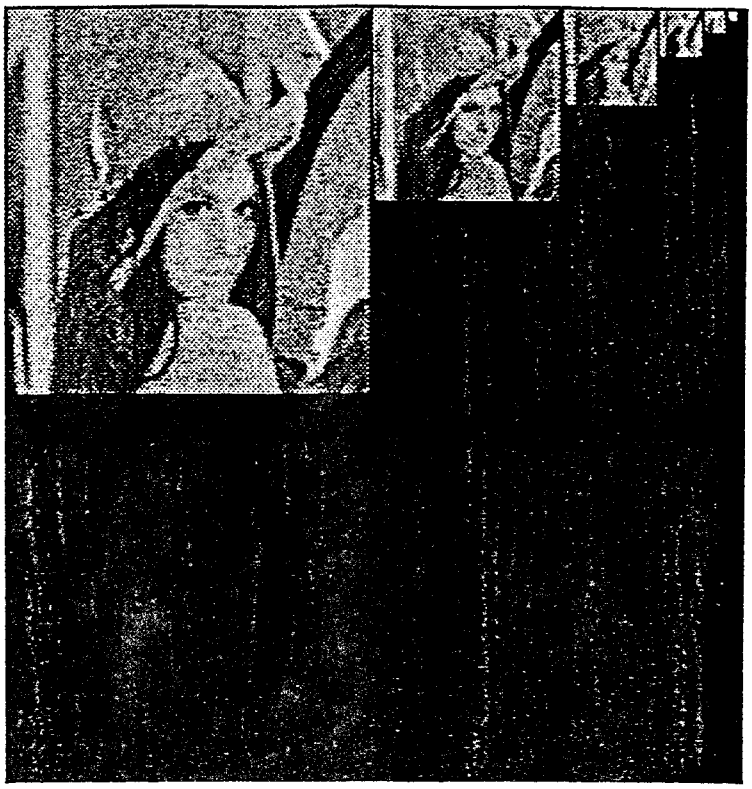

Figure 4: Low-pass pyramid representation of the 'girl' image (levels 1 to 6)

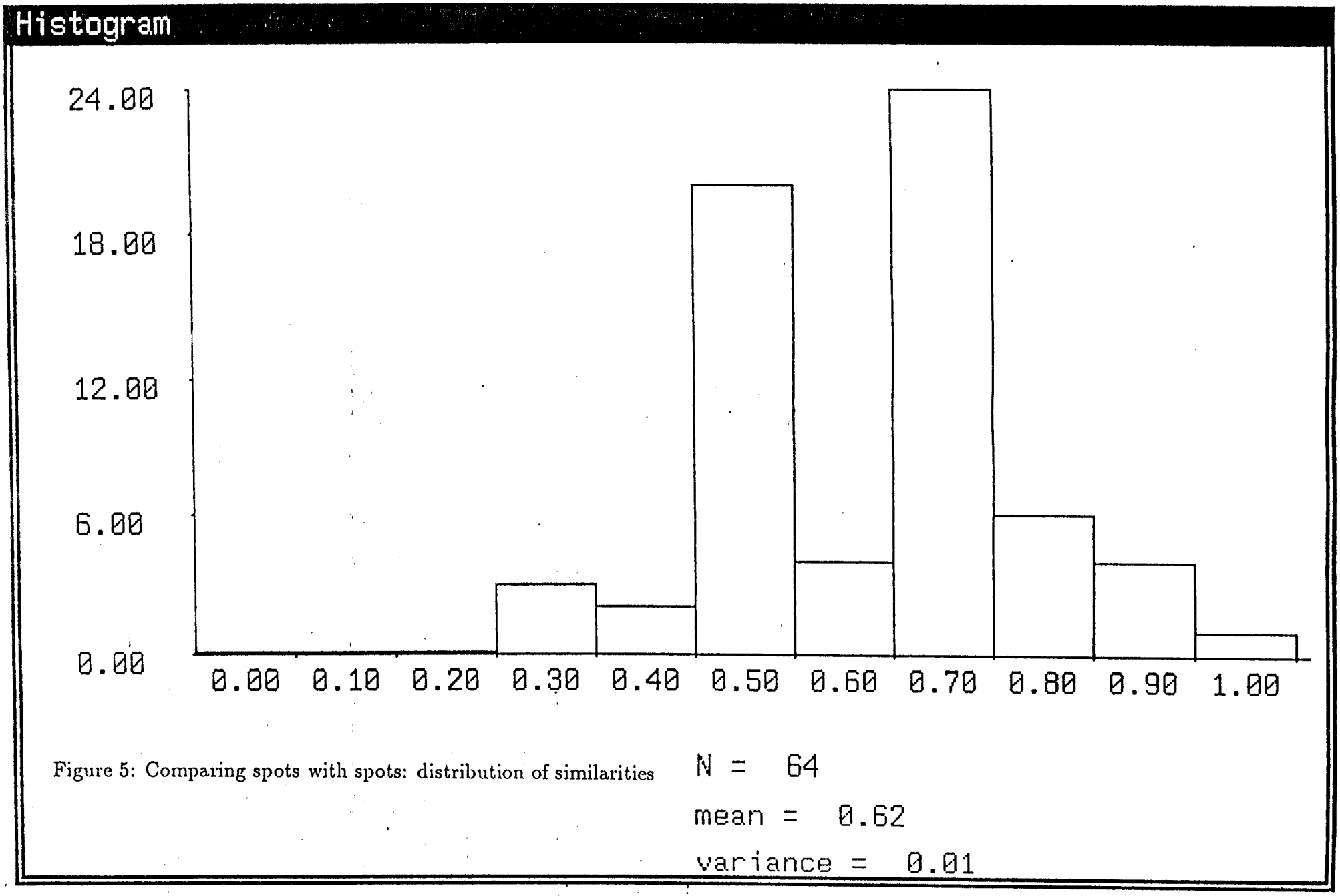

Figure 5: Comparing spots with spots: distribution of similarities

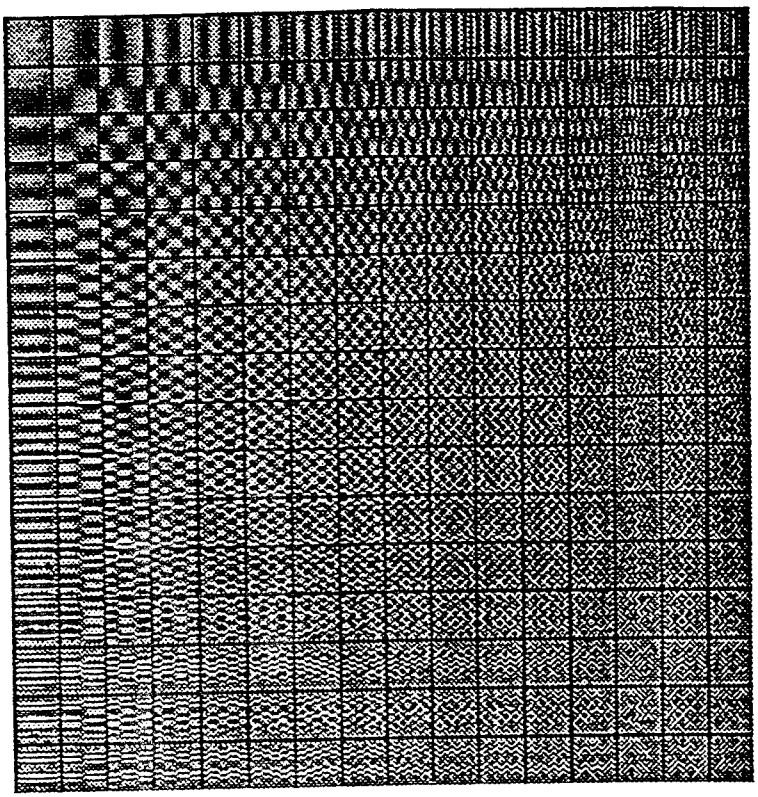

Figure 8: Level 1 eigenpatterns

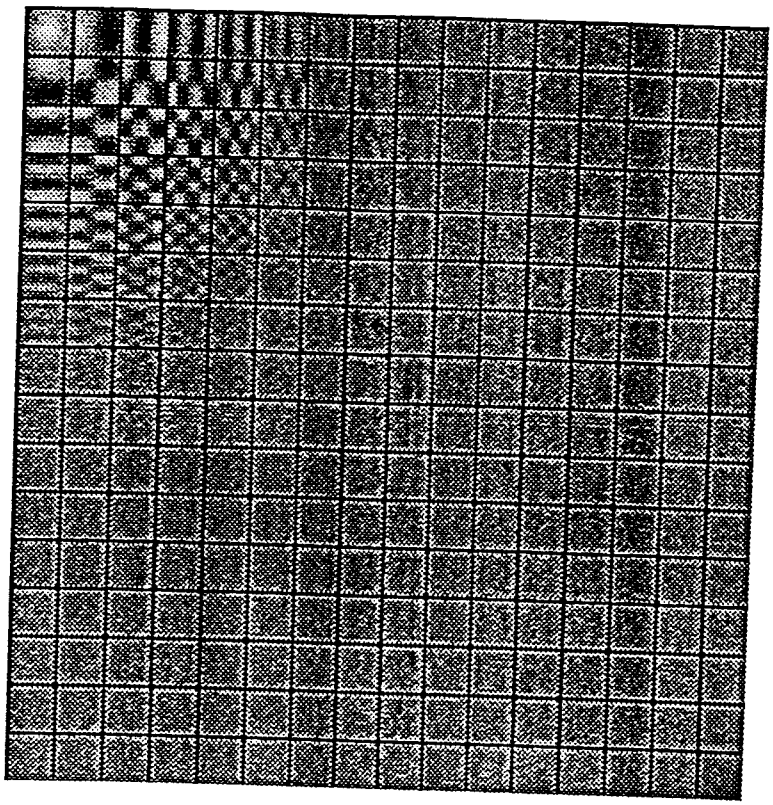

Figure 9: Level 1 eigenpatterns weighted by their eigenvalues

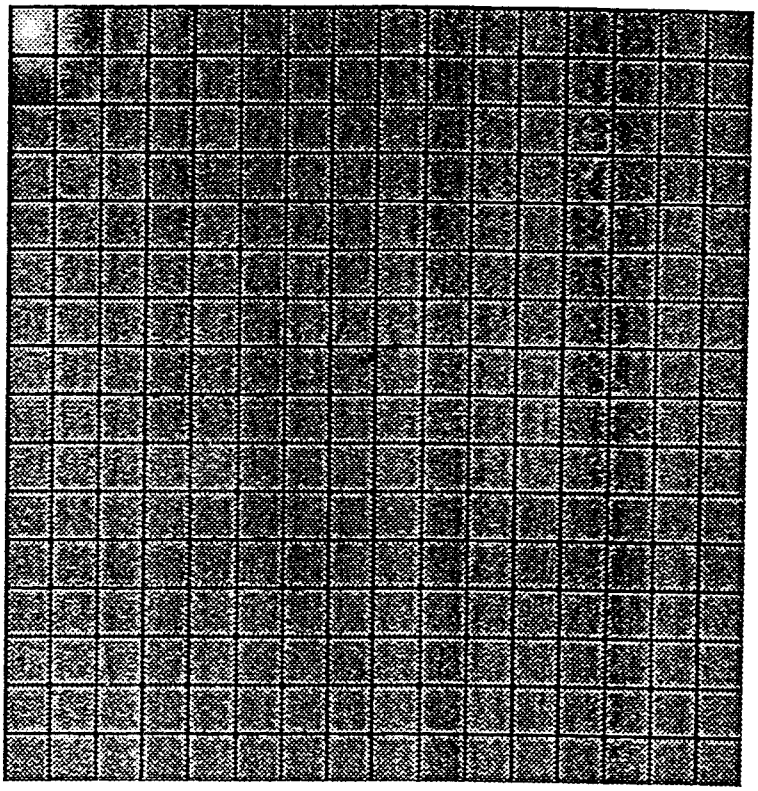

Figure 10: Level 3 eigenpatterns weighted by their eigenvalues

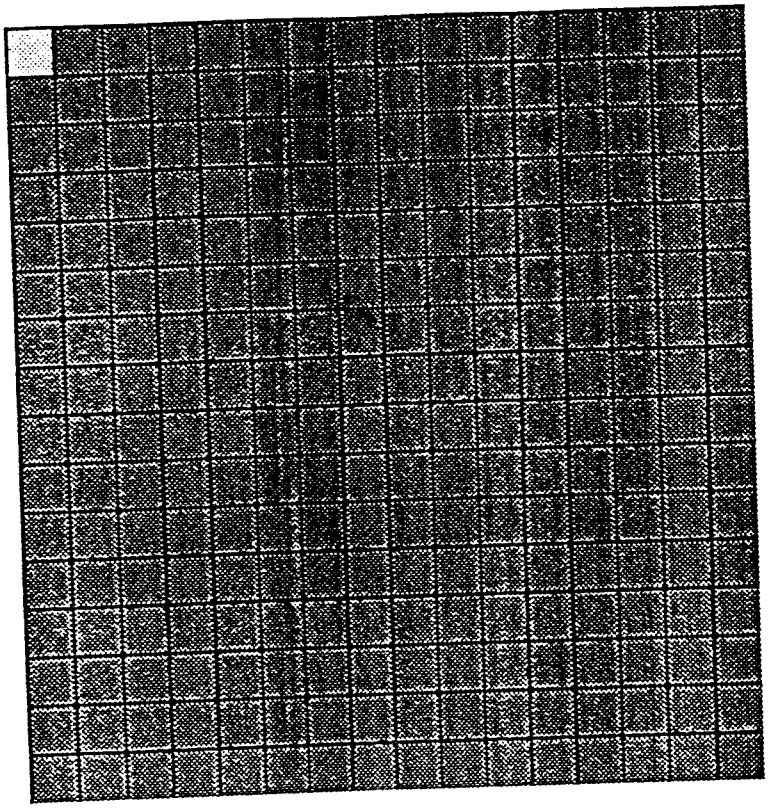

Figure 11: Level 6 eigenpatterns weighted by their eigenvalues

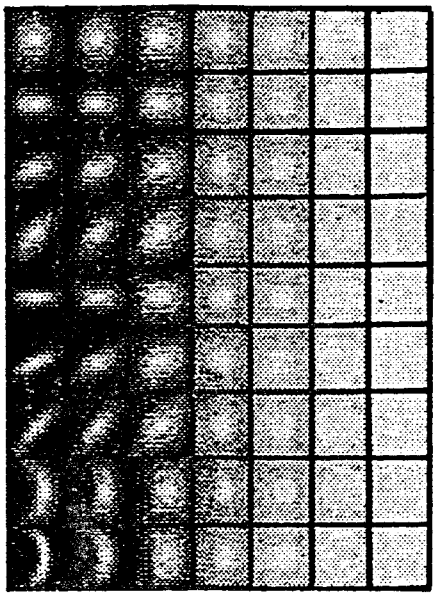

Figure 12: Oriented Gaussian patterns (L to R: original 'blobs'; Levels 1 - 6 of low-pass pyramid)

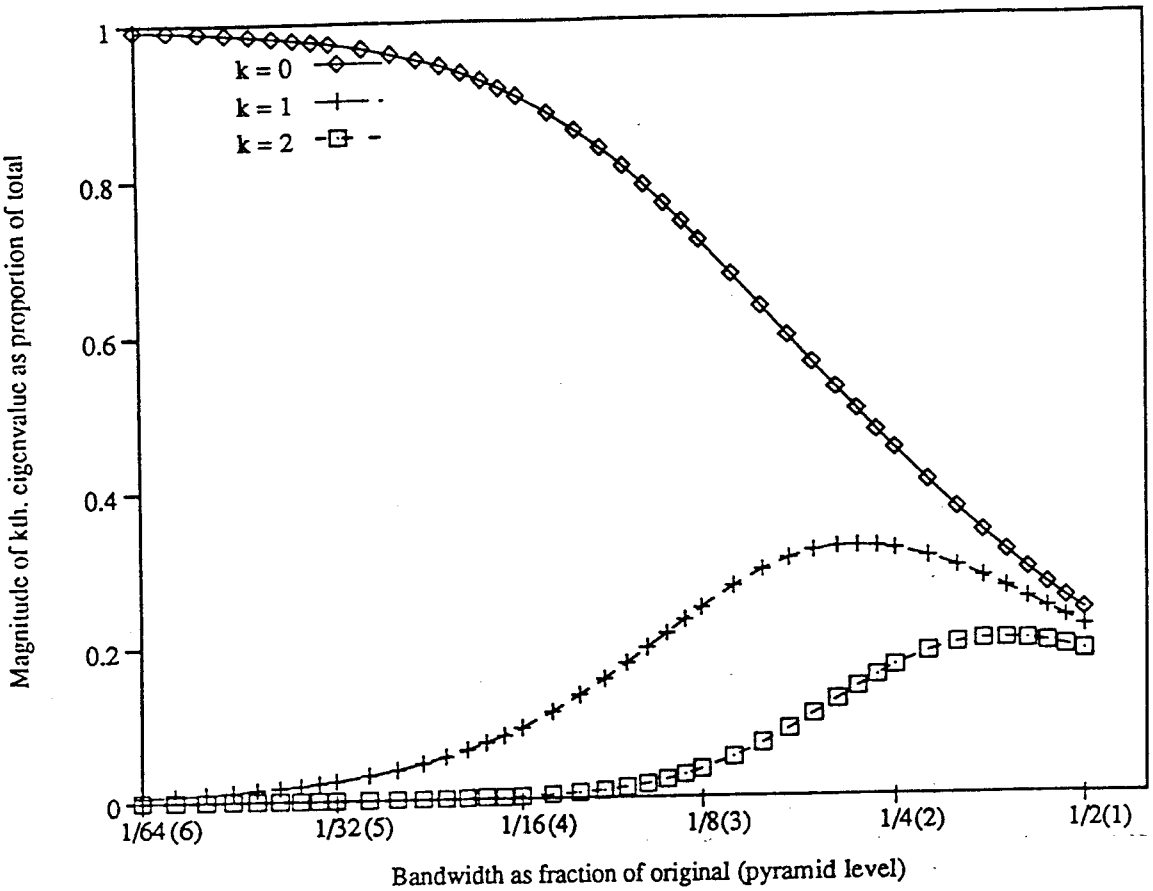

Figure 13: Magnitudes of first three eigenvalues

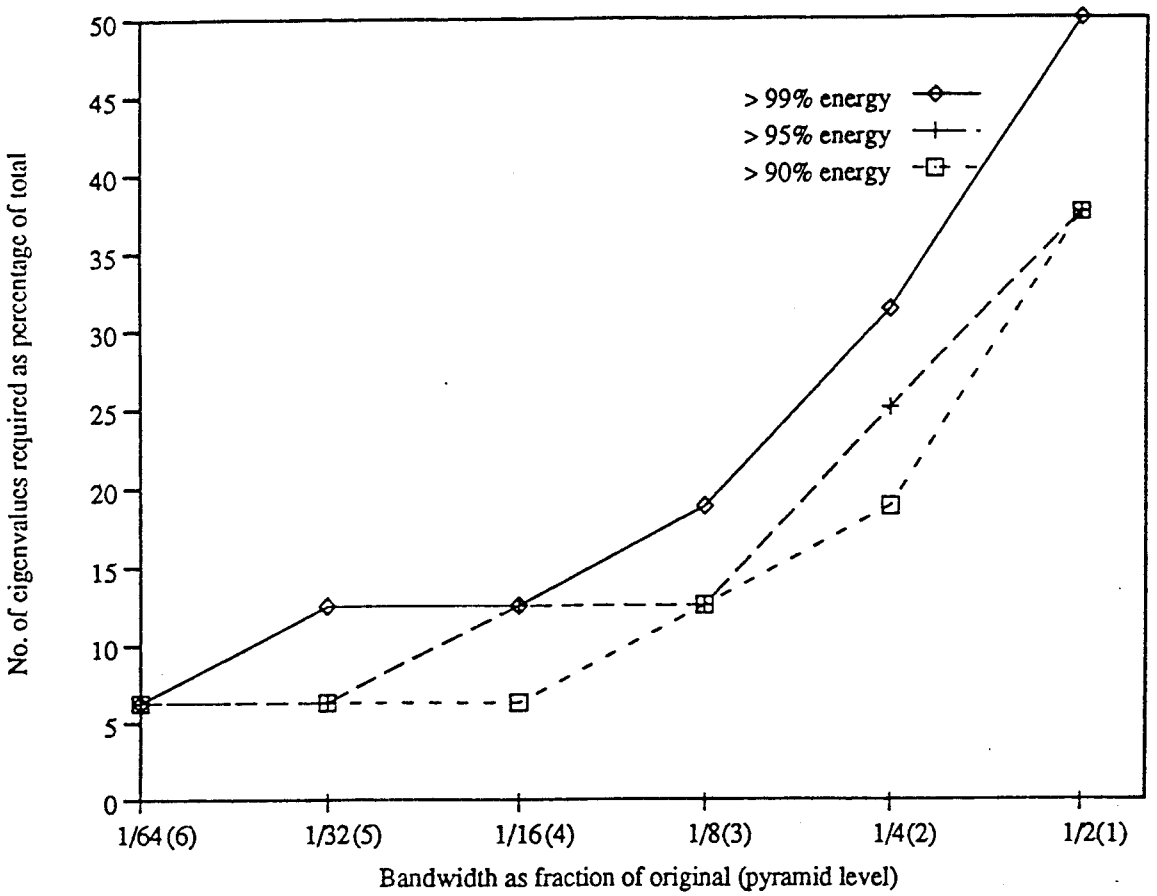

Figure 14: Amount of energy retained in first few eigenvectors

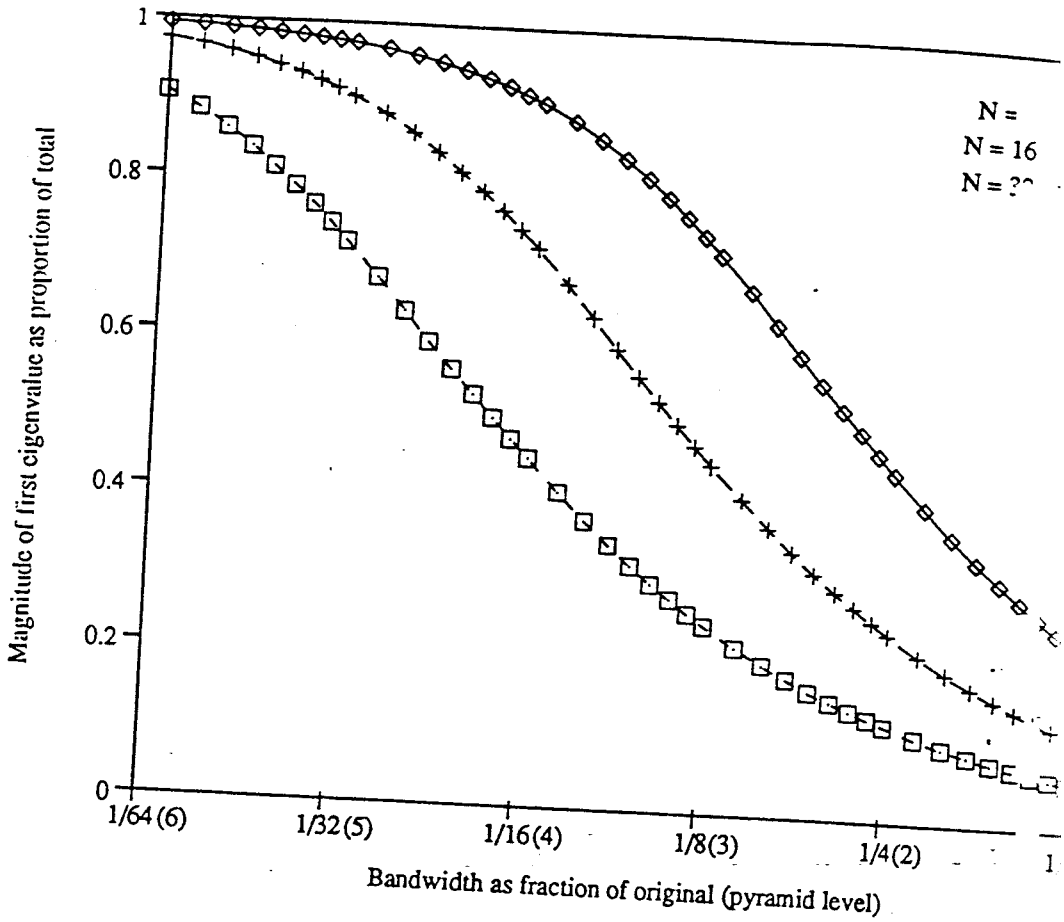

Figure 15: Magnitudes of λ_0 for N = 8, 16, 32

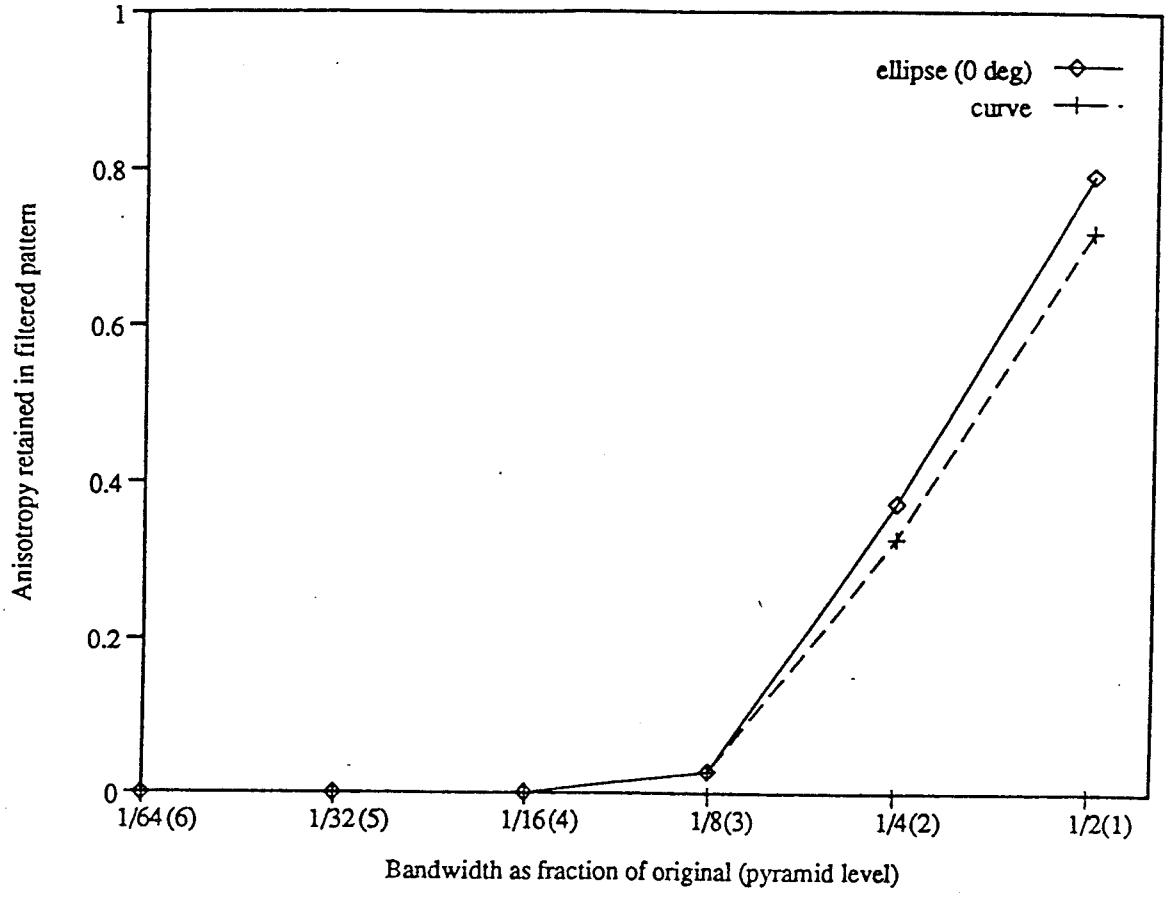

Figure 16: Preservation of anisotropy for Gaussian ellipse and curve

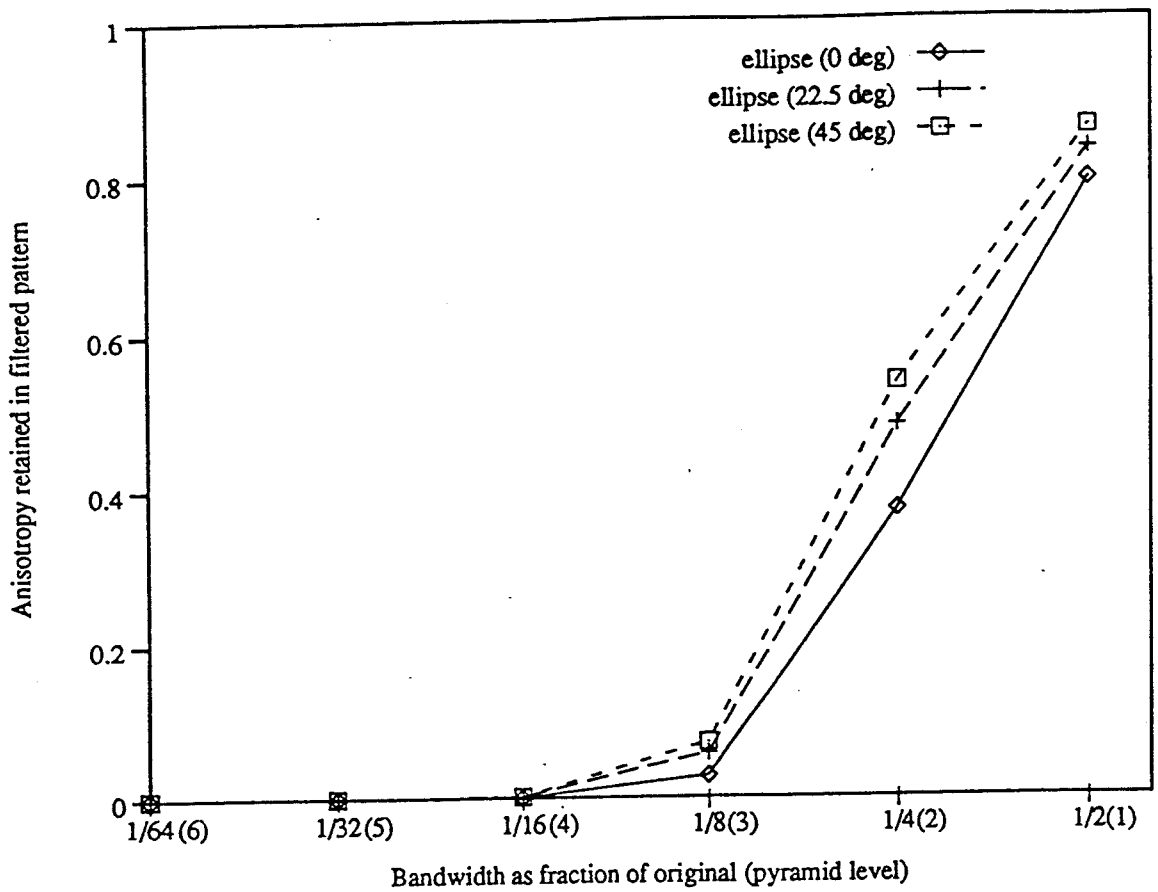

Figure 17: Preservation of anisotropy for Gaussian ellipses at various orientations

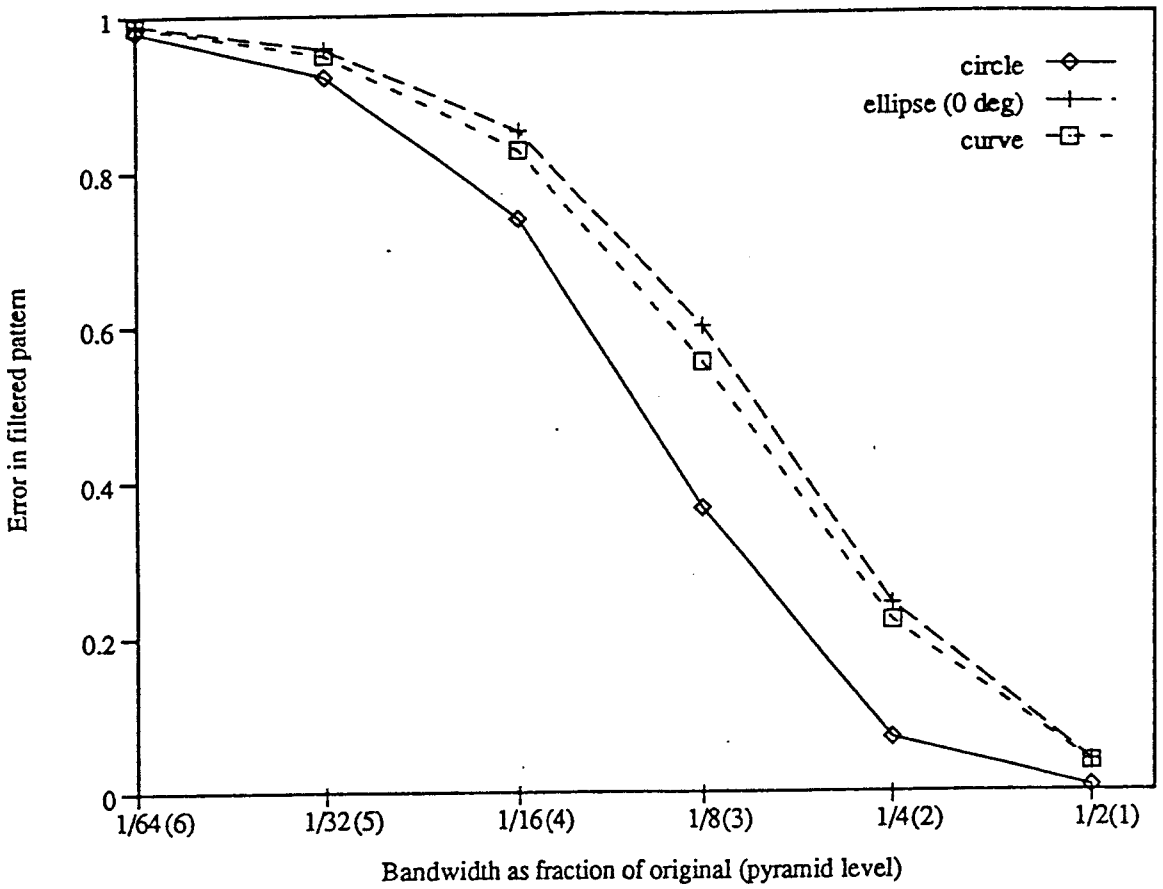

Figure 18: Errors in various Gaussian blobs for N = 8