Feature grouping by "relocalisation" of eigenvectors of the proximity matrix

Guy L. Scott
Robotics Research Group
Department of Engineering Science
University of Oxford

H. Christopher Longuet-Higgins
University of Sussex
Falmer
Brighton

We describe a widely applicable method of grouping - or clustering - image features (such as points, lines, corners, flow vectors and the like). It takes as input a "proximity matrix" **H** *- a square, symmetric matrix of dimension* N *(where* N *is the number of features). The element* i,j *of* **H** *is an initial estimate of the "proximity" between the* i-*th and* j-*th features. As output it delivers another square symmetric matrix* **S** *whose* i-j*th element is near to, or much less than unity according as features* i *and* j *are to be assigned to the same or different clusters.*

To find **S** *we first determine the eigenvalues and eigenvectors of* **H** *and re-express the features as linear combinations of a limited number of these eigenvectors - those with the largest eigenvalues. The cosines between the resulting vectors are the elements of* **S**. *We demonstrate the application of the method to a range of examples and briefly discuss various theoretical and computational issues.*

In studying various problems in computer vision we have hit upon an apparently novel method of cluster analysis [4] related to a technique widely used in molecular physics [2] [6]. The input to the method is a matrix **H** of pairwise proximities in, for example, a two dimensional image; the output is closely related to the molecular concept of a "bond order" matrix which indicates whether any two features do or do not belong to the same cluster.

Let us consider a feature space (2-dimensional for simplicity) containing three features A, B, C as shown in figure 1.

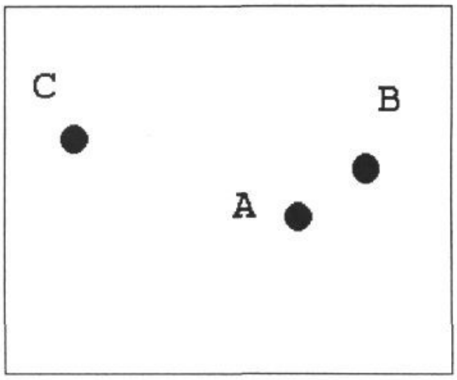

Figure 1: Three-point configuration

Let us define as the measure of proximity between any pair of features the quantity $\exp(-d^2/2\sigma^2)$ where d is the Euclidean distance beteen the two features and σ is a scale constant. For a value of σ approximately equal to the distance between A and B the proximity matrix **H**, shown in figure 1, is obtained.

	A	B	C
A	1.00	0.63	0.03
B	0.63	1.00	0.0
C	0.03	0.0	1.00

Table 1.

We wish to analyse this matrix to obtain a clustering; that is, we wish to automatically extract one "cluster" which includes only A and B and another that includes only C. In this 3-feature case there are many trivial procedures that might be applied, but we desire a method that will generalise to proximity matrices of arbitrary size and complexity.

To analyse the connectivity implicit in **H** effectively we need to extract a fair number of eigenvectors. In the 3×3 example given above the eigenvalues and eigenvectors are as follows:

	E_1	E_2	E_3
Eigenvalues	1.63	1.00	0.37
A	-0.71	-0.01	0.71
B	-0.71	-0.05	-0.71
C	-0.04	1.00	-0.03

Table 2.

The eigenvector E_1 with the highest eigenvalue (that is to say the eigenvector that "does the most to explain" the structure of **H**) consists of almost equal parts of A and B - and very little of C. (The sense of an eigenvector is arbitrary - only its direction counts - so the components could all be reversed in sign). The second eigenvector E_2 consists almost exclusively of C. Like E_1, E_3 consists of A and B with very little C but now the components associated with A and B are *opposite in sign*. By analogy with the analysis of modes of vibration we shall sometimes refer to the eigenvectors as "modes".

We might feel inclined to regard our eigenvector decomposition of **H** as, to all intents and purposes, a clustering. The first two vectors cleanly divide the three features into two groups along the desired lines. However, our clean result has been achieved by contrivance. A suitable choice of image and of the scale constant σ ensured the fulfilment of conditions that in general we cannot expect to hold. First, the two "groups" - AB and C in its solitary splendour - are almost disjoint in the sense of being linked by very small elements of the matrix **H**. Also, the three eigenvalues are well separated. There is thus no degeneracy causing arbitrary linear mixing of the desired modes.

These two conditions ensure no "contamination" of one group by another. But if we enlarge σ to a value closer to AB than AC – to obtain a stronger association between C and the other two features – we obtain the following **H**:

	A	B	C
A	1.00	0.91	0.51
B	0.91	1.00	0.34
C	0.51	0.34	1.00

Table 3.

The eigenvalues and vectors are now as follows:

	E_1	E_2	E_3
Eigenvalues	2.21	0.72	0.07
A	0.65	-0.19	0.73
B	0.61	-0.43	-0.66
C	0.44	0.88	-0.16

Table 4.

It is clear from inspection of this table that our features are no longer cleanly segmented by the eigenvector decomposition. The non-neglible interaction of C with A and with B has "pulled" the first mode towards it.

"RELOCALISATION"

The *columns* in table 4 above define the eigenvectors (or modes) as linear combinations of features. The *rows* represent the "coordinates" of each feature measured along the various eigenvectors - they represent the "expansion" of that feature in terms of the eigenvector basis. Let us call the whole row associated with a feature its F-vector and denote it by F_i where i labels the feature. For example F_B represents the row labelled B in table 4. Note that the F-vectors of any two distinct features are necessarily orthogonal, that is $F_i \cdot F_j = 0$ for differing i and j. But we will be less concerned with F_i than with its "truncated" form T_i, in which the components associated with the eigenvectors beyond a certain point in the eigenvalue-ordered sequence are set to zero. We will use M to denote the number of modes up to the cut-off point.

With M=2 we have, in the case described by table 4, $T_A = [0.65, -0.19, 0]$ and $T_B = [0.61, -0.43, 0]$. They are not orthogonal at all. In fact they are quite similar. By contrast, they both differ markedly from T_C, being roughly (but not exactly) orthogonal to it. We now compute a "bond order" matrix **P** whose element P_{ij} is the *scalar product* of the i-th and j-th T-vectors:

	A	B	C
A	0.46	0.48	0.12
B	0.48	0.57	-0.11
C	0.12	-0.11	0.97

Table 5.

Note that the 2×2 "block" subtended by features A and B has elements approximately 0.5 in magnitude, while the "1×1 " block corresponding to the solitary feature C has magnitude close to 1.0. It is in fact possible to show that if the features fall into well separated groups, and the number of eigenvectors used in the reconstruction is equal to the number of groups, then every element in an $n \times n$ block will be equal to 1/n.

The association indicated by the closeness of T_A and T_B can be made more apparent - both to human and machine - if we compute an "association matrix" **S** whose element S_{ij} is the *cosine of the angle* between the i-th and j-th T-vectors:

	A	B	C
A	1.00	0.95	0.18
B	0.95	1.00	-0.15
C	0.18	-0.15	1.00

Table 6.

Comparison of table 6 with table 3 reveals a marked change in the direction of "high contrast" (with matrix entries tending toward either unity or zero).

THE GENERAL CASE

We use subscripts i and j to refer to *features* ($i, j \leq N$), and r and s ($r, s \leq N$) to denote eigenvalues/vectors.

The proximity of features i and j is denoted by H_{ij} ($= H_{ji}$). The *proximity matrix* **H** has eigenvaues λ_r and eigenvectors E_r satisfying

$$\mathbf{H}E_r = \lambda_r E_r \tag{1}$$

The eigenvalues λ_r are ordered in decreasing sequence

$$\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_N$$

We denote by **V** the matrix whose columns are the eigenvectors $E_1, E_2, ..., E_N$. The rows of **V**

$$F_i = [V_{i1}, V_{i2}, ..., V_{iN}]$$

correspond to the features. Provided the eigenvectors E_r are properly normalised (and rendered orthogonal in the case of degeneracy) **V** will be an orthogonal matrix, satisfying

$$\mathbf{V}\mathbf{V}^{\mathbf{T}} = \mathbf{I} = \mathbf{V}^{\mathbf{T}}\mathbf{V} \tag{2}$$

where **I** is the identity matrix. If **D** is the diagonal matrix

$$\mathbf{D} = diag[\lambda_1, \lambda_2, ..., \lambda_N]$$

then $\mathbf{V}^{\mathbf{T}}\mathbf{H}\mathbf{V} = \mathbf{D}$ and $\mathbf{V}\mathbf{D}\mathbf{V}^{\mathbf{T}} = \mathbf{H}$.

Truncation of F_i after M components gives

$$T_i = [V_{i1}, ..., V_{iM}, 0, ..., 0]$$

The matrix whose rows are T_i is denoted by **W**.

Let **J** be the diagonal matrix whose first M components diagonal elements are 1, the others being 0. Then:

$$\mathbf{P} = \mathbf{V}^{\mathbf{T}}\mathbf{J}\mathbf{V}$$

is the "bond order" matrix (the language is that of molecular physics); its elements satisfy

$$P_{ij} = \sum_{r=1}^{M} V_{ir} V_{jr} = \sum_{r=1}^{N} W_{ir} W_{jr} \tag{3}$$

in which the first sum is restricted, the second not.

The association matrix **S** has elements

$$S_{ij} = \frac{\sum_r W_{ir} W_{jr}}{\sqrt{\sum_r W_{ir}^2}\sqrt{\sum_r W_{jr}^2}} \tag{4}$$

The entry P_{ij} in the bond order matrix is the scalar product of the truncated feature vectors T_i and T_j. These vectors in general have magnitude less than unity, and so the elements of the bond order matrix can give the impression of a low apparent degree of association between two features, despite their T-vectors being similar or even identical. For this reason the association matrix **S** is formed by first normalising the T-vectors so that they have unit magnitude and then forming the scalar products. The entry S_{ij} is thus the cosine of the *angle* between the i-th and j-th T-vectors, and is independent of their magnitude.

Like **P**, **S** may contain negative elements, one indication that we have not got a very clear-cut partitioning. If so, the situation can sometimes be improved by the following procedure:

1. Return to the *bond order* matrix **P** - the unnormalised version of **S**.
2. Set the negative entries in **P** to zero, to obtain a new matrix $\mathbf{H}^*$.
3. Using $\mathbf{H}^*$ as proximity matrix apply the original procedure to obtain a new bond order matrix $\mathbf{P}^*$, say.
4. Repeat steps 2-3 until negative entries in the bond order matrix are insignificant by some criterion.
5. Re-compute the association matrix **S**.

In molecular physics negative bond orders correspond to repulsive electrodynamic forces between two atoms. Our iterative procedure may be regarded - very loosely! - as accommodating "repulsion" by moving features apart so that their proximity approaches zero.

IMAGE CLUSTERING – AN EXAMPLE

Figure 2 shows a dot-pattern extracted from the wing-markings of the common southern African butterfly *Hamanumida Daedalus.* Figures 3 and 4 show the promixity matrix **H** for two widely separated values of σ (the ratio between them is 1:10). For the purposes of display only we have ordered the points in such a way as to obtain an **H** matrix that is as "block-like" as possible. Note that the ordering of features makes no difference whatsoever to the logic of our method. Figure 3 shows **H** for a "short-range" σ, considerably smaller than the separation between the left- and right-hand wing-pairs. Figure 4

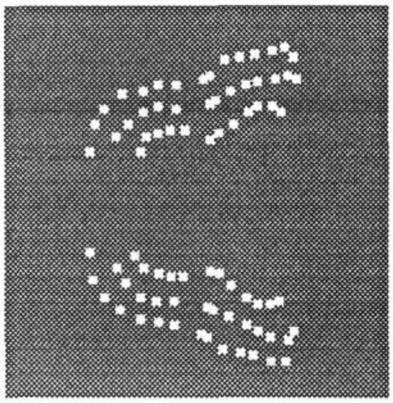

Figure 2: Dot-pattern from butterfly wings

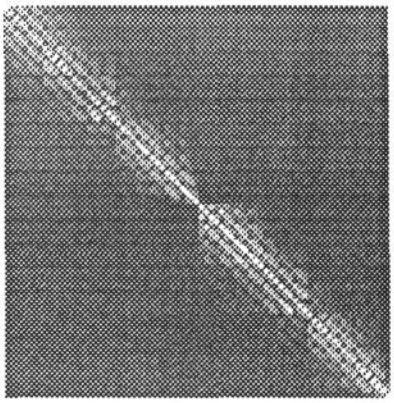

Figure 3: "Short-range" proximity matrix for butterfly

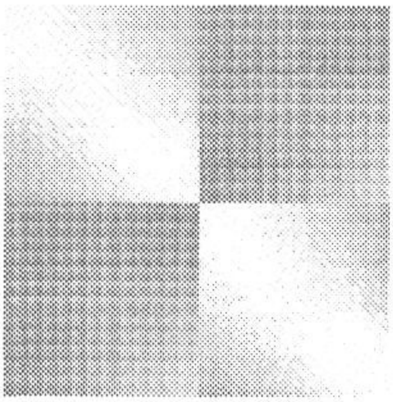

Figure 4: "Long-range" proximity matrix for butterfly

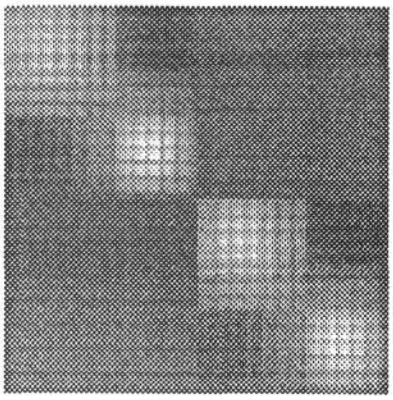

Figure 5: **P** matrix derived from matrix of figure 3

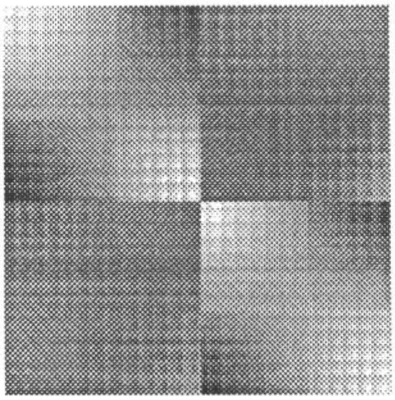

Figure 6: **P** matrix derived from matrix of figure 4

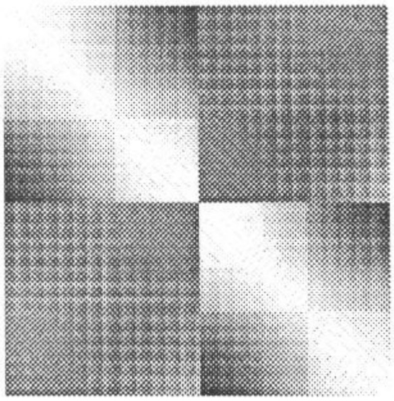

Figure 8: **S** matrix derived from matrix of figure 4

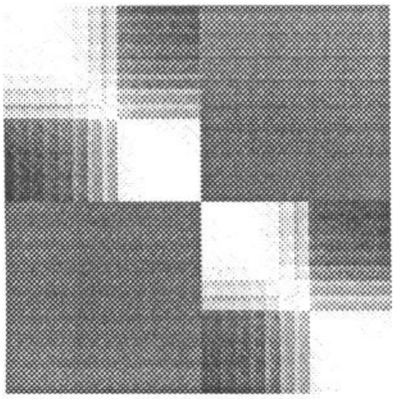

Figure 7: **S** matrix derived from matrix of figure 3

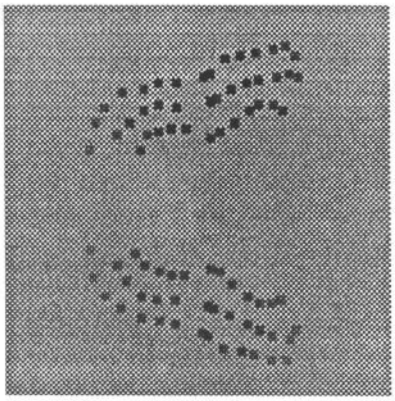

Figure 9: First mode of the butterfly pattern

shows **H** for a "long-range" σ in which proximity between points on opposite sides is considerable.

Figures 5 and 6 show the **P** matrices resulting, for the short- and long-range σ respectively, from one pass of our procedure with M=4. Note the considerable enhancement of the block structure. In order from the top left the four "blocks" correspond to: the fore-left, the hind-left, the fore-right and the hind-right wings. Also note the presence of negative (i.e. darker) regions corresponding to "repulsion" between the fore- and hind-wings on each side.

Figures 7 and 8 show the **S** matrices corresponding to 5 and 6 respectively. The variation in intensity in figures 5 and 6 which is due to variation in the *amplitude* of the T-vectors has been removed by the normalisation process and the result is a higher-contrast image. Note also that intensity scaling of the grey-level image for display disguises the fact that the intra-block entries in figures 5 and 6 are of typical amplitude 0.05 (there being abour 20 points per block) whereas those in figures 7 and 8 are of amplitude close to 1.0.

The similarity of the results obtained with very different values of σ demonstrates the robustness of our method to this parameter - at least in the case where there truly *are* distinct clusters of features.

To generate the remaining figures referred to in this section we employed an "intermediate" value of σ. The first 10 eigenvalues of **H** are:

22.31 20.95 10.91 9.79 3.39 2.91 2.43 1.92 0.99 0.88

The closeness of eigenvalues 1 and 2 – and of 3 and 4 – is diagnostic of "near-degeneracy" and "contamination". Inspection of the first four modes (figures 9 to 12 inclusive) reveals the sort of delocalisation we encountered in the simple 3-feature case. These images are obtained by plotting, *at* the geographical location of each butterfly-point i, a square with a grey-level corresponding to the value of V_{ir} – where r indexes the mode being displayed.

We first set M=2 and construct a association matrix **S**. The i-th row of this can be regarded as a list of revised "association strengths" of feature i with all features. Many rows are, of course, rather similar to other rows (this is what it means for a matrix to have a block-like structure). To obtain figures 13 and 14 we selected two representative rows of **S** - those corresponding to the features marked with a cross - and plotted them exactly as we plotted the modes in the preceding figures. We find that we have "lit up" the left and right wing-pairs separately.

Increasing M to 4 and repeating the procedure just described we obtain four distinct groups (there now being four "blocks" in **S**). Two of these are shown in figures 15 and 16 (the other two are mirror images grouping points on the right hand wing). In these figures the negative associations with the "crossed" features are more prominent than in figures 13 and 14, where they are barely visible. Iteration in the manner described in the previous section is found to reduce their magnitudes considerably.

An obvious question that arises concerns the "optimal" choice of M - the effective dimension of the T-vectors. First, it is plainly a bad idea to truncate the mode se-

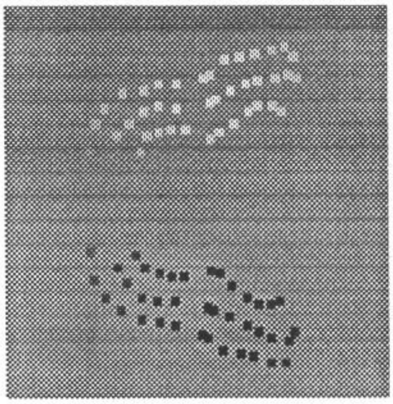

Figure 10: Second mode of the butterfly pattern

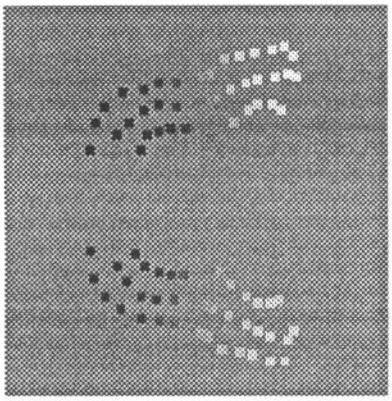

Figure 11: Third mode of the butterfly pattern

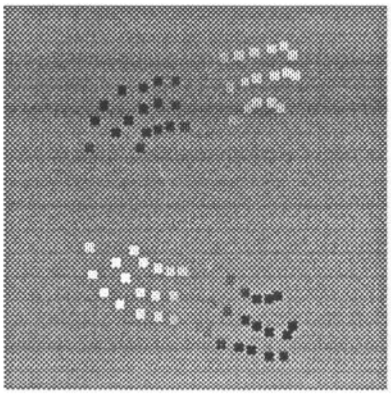

Figure 12: Fourth mode of the butterfly pattern

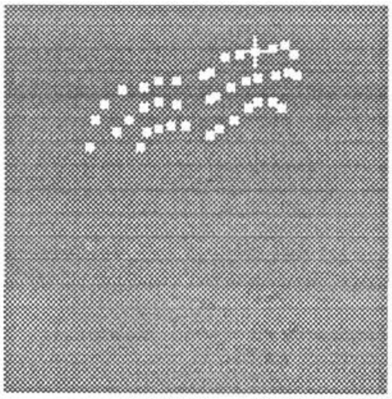

Figure 13: "Group" derived with M=2

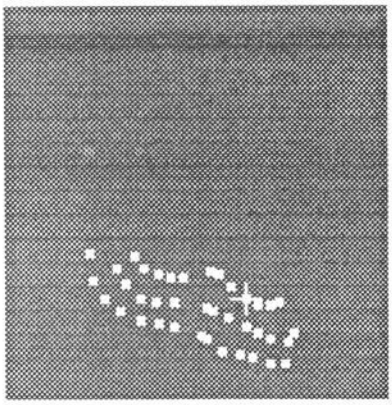

Figure 14: Second "group" derived with M=2

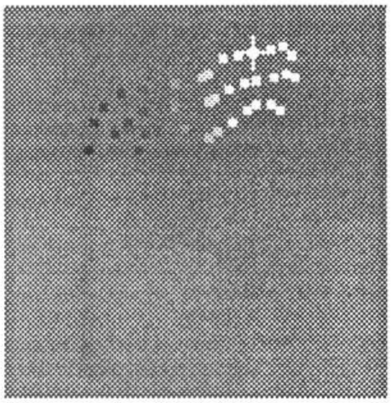

Figure 15: "Group" derived with M=4

quence between two nearly equal eigenvalues since the result would then be affected by small changes in the data. For example mode number M (included) and mode number $M+1$ (excluded) could be made to swap round with a small change of σ or small perturbations of features. To avoid this, there should be a marked step in "energy" between the last included and the first excluded mode.

Second, it is not clear that a *single* value of M is what is required for a grouping analysis. We might naturally want to analyse our image hierarchically [4]; first at M=2 to obtain a left-wing/right-wing segmentation and then at M=4 to obtain a front-wing/back-wing segmentation. But we are very far from having solved the problem of the automatic selection of a value (or values) for M.

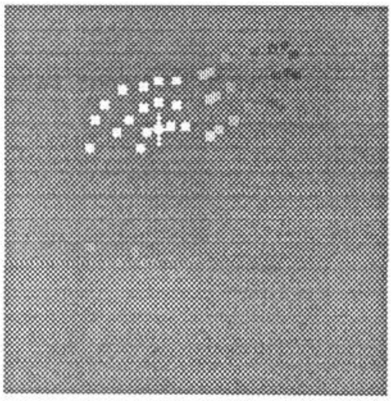

Figure 16: "Group" derived with M=4

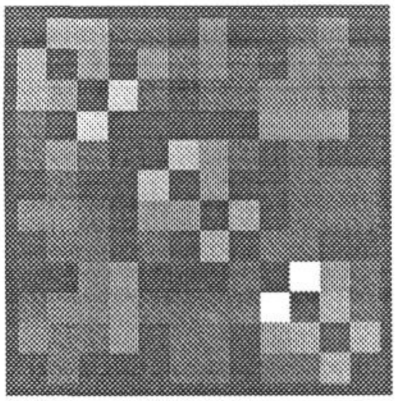

Figure 17: Initial proximity data as provided by Cowie

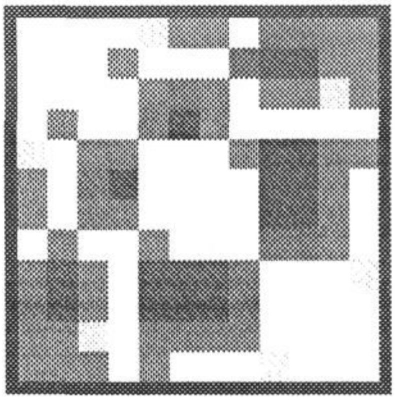

Figure 18: Output of Cowie's program

GROUPING IN A COMPLEX FEATURE SPACE

At the 1989 AVC conference R. Cowie et al [3] presented a paper that described a way of grouping the vertices in a line-drawing, by first constructing a proximity matrix, and then deriving a clustering from it. Unfortunately, the results of the second stage left something to be desired, so we have applied our own clustering method to the Belfast proximity data, kindly made available to us by Dr. Cowie.

Figure 17 shows one of Cowie's proximity matrices (for the Penrose triangle) in a grey level representation. This corresponds to **H** in our nomenclature. Figure 18 shows a result of application of Cowie's program. Figure 19 shows the association matrix **S** resulting from one pass with our program. Figure 20 shows **S** after iterating five times with our program for the same value of **M**. The block structure of the resulting association matrix is much clearer than in figure 19, and one of us (GLS) is collaborating with Dr. Cowie on further applications of our method.

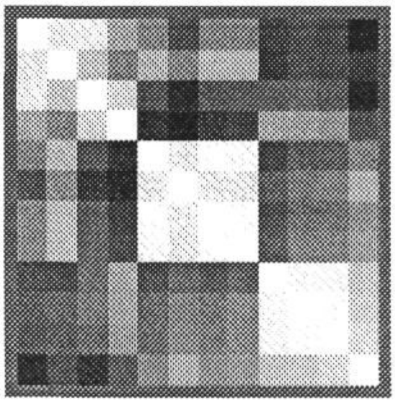

Figure 19: Association matrix **S** - no iteration

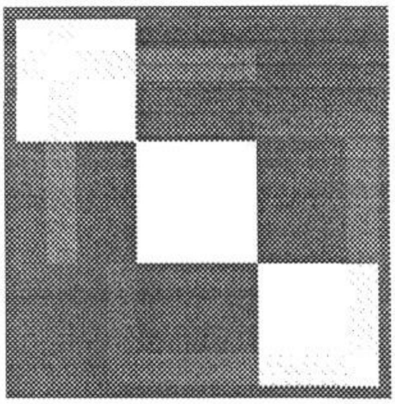

Figure 20: Association matrix **S** - after 5 iterations

A NOTE ON COMPUTATION

Accurate and complete solutions to eigenvalue problems tend to be computationally expensive. The time scales as N^3, being about 25 seconds on a Sun 3 for $N = 79$. Leaving aside the undoubted possibilities for parallelisation and analogue computation, we note that time may be considerably reduced if advantage is taken of a number of factors including: the fact that a proximity matrix is often very sparse; not all eigenvalues and vectors are needed (only as many as there are clusters); and the required accuracy is low (we do not care if eigenvectors of similar frequency are "mixed").

We have also considered extending our method to the domain of continuous image functions using basis functions such as a fixed grid of Gaussians or Fourier components. These extensions are to be presented and discussed elsewhere.

References

[1] **Brady J M & Scott G L** "Parallel Algorithms for Shape Representation" in **Page I** *Parallel Architectures and Computer Vision* Clarendon: Oxford (1988)

[2] **Coulson C A** *Valence.* 2nd edition Oxford University Press (1961) eps pp 268-270.

[3] **Cowie R.I.D, Hamill T, Morrow P & Perrott R H** "Interpreting Line Drawings Using a Clustering Technique" *Proc. 5th AVC* (1989) pp 37-42

[4] **Everitt B** *Cluster Analysis.* London: Heinemann (1974).

[5] **Heine V** *Group theory in quantum mechanics.* Pergammon Oxford 1960

[6] **Longuet-Higgins H C & Salem L** " The Alternation of Bond-length in Long Conjugated Chain Molecules" *Proc. Roy. Soc. A***251** pp 172-185 (esp. 174)