Overcoming the Curse of Dimensionality in Clustering by means of the Wavelet Transform

Fionn Murtagh
School of Computer Science, The Queen's University of Belfast,
Belfast BT7 1NN, Northern Ireland

Jean-Luc Starck
CEA, SEI/DSM/DAPNIA, CE-Saclay, F-91191 Gif-sur-Yvette Cedex, France

Michael W. Berry
Ayres Hall 114, Department of Computer Science,
University of Tennessee, TN 37996-1301

The Computer Journal, 43, 107-120, 2000

Abstract

We use a redundant wavelet transform analysis to detect clusters in high-dimensional data spaces. We overcome Bellman's "curse of dimensionality" in such problems by (i) using some canonical ordering of observation and variable (document and term) dimensions in our data, (ii) applying a wavelet transform to such canonically ordered data, (iii) modeling the noise in wavelet space, (iv) defining significant component parts of the data as opposed to insignificant or noisy component parts, and (v) reading off the resultant clusters. The overall complexity of this innovative approach is linear in the data dimensionality. We describe a number of examples and test cases, including the clustering of high-dimensional hypertext data.

1 Introduction

Bellman's (1961) [1] "curse of dimensionality" refers to the exponential growth of hypervolume as a function of dimensionality. All problems become tougher as the dimensionality increases. Nowhere is this more evident than in problems related to search in high-dimensional information spaces. The final goal is information resource discovery or finding, the navigation of the user search process is a primary means towards this end, and clustering or other structuring is often a prerequisite for this. Clustering is part and parcel of information retrieval whether it is algorithmic, and seeks to expedite the search process, or is cognitive and related to the user's understanding and intention.

In [2] (see also [3]), a constant computational time or $O(1)$ approach to cluster analysis was described. The computational complexity was, as is usual, defined in terms of the number of observations. This work related to problem spaces of dimensionality 2, with generalization possible to 3-dimensional spaces [4]. It may be helpful to distinguish this work from clustering understood as mixture distribution modeling. Banfield and Raftery [5], for example, discuss algorithms for optimal cluster modeling and fitting. Murtagh and Starck's [2] work on $O(1)$ clustering algorithms is based on noise modeling. It can accurately be defined as data background modeling.

In this article we describe an effective approach for clustering high-dimensional data. Many wavelet transforms are of $O(n)$ computational complexity for an n-length input data stream. Examples include such orthonormal transforms as the Haar and Daubechies transforms (for a very readable and well-based mathematical overview see [6]) and the redundant à trous transform used below. In terms of the space dimensionality, therefore, the computational complexity is $O(n)$, or linear. The computational complexity is independent of the number of items that we are studying and hence this is $O(1)$ or constant-time. Such "items" could be document-term co-occurrences, or existence of hyperlinks. All other processing we carry out is local and is dominated by the aforementioned computational complexity.

The "trick" we use to achieve these breakthrough results is derived from data visualization. We take the document-term or hyperlink array as a 2-dimensional image. In general, an array is a mapping from the Cartesian product of observation set, I, and attribute set, J, onto the reals, $f : I \times J \longrightarrow \mathbb{R}$, while an image (single frame) is generally defined for discrete spatial intervals X and Y, $f : X \times Y \longrightarrow \mathbb{R}$. A table or array differs from a 2-dimensional image, however, in one respect. There is an order relation defined on the row- and column-dimensions in the case of the image. To achieve invariance we must induce an analogous ordering relation on the observation and variable dimensions of our data table.

A natural way to do this is to seek to optimize contiguous placement of large (or nonzero) data table entries. Note that array row and column permutation to achieve such an optimal or suboptimal result leaves intact each value x_{ij}. We simply have row and column, i and j, in different locations at output compared to input. Methods for achieving such block clustering of data arrays include combinatorial optimization ([7, 8, 9]) and iterative methods ([10, 11]). In an information retrieval context, a simulated annealing approach was also used in [12]. Further references and discussion of these methods can be found in [13, 14, 15]. Treating the results of such methods as an image for visualization purposes is a very common practice (e.g. [16]).

A further class of methods has been studied by [17]. This work is based on matrix reordering schemes such as the Reverse Cuthill-McKee method. These matrix reordering schemes are used to diagonalize large sparse matrices in the context of analysis methods such as correspondence analysis and latent semantic indexing. The matrix reordering schemes themselves may be very fast. Berry et al. [17] discuss such a method which, in the case of a very sparse matrix, is of computational complexity proportional to the number of nonzero values in the matrix. Other methods discussed include fast sparse matrix reorderings related to correspondence analysis. Such a reordering method is used in the last example, that of the encyclopedia data, discussed below.

2 Incidence Data and Wavelet Transforms

Consider co-occurrence data, or document-term dependence data. Contiguity of links, or of data values in general, is important if we take the 2-way data array as a 2-dimensional image. It is precisely this issue which distinguishes a data array from an image: in the latter data type, the rows and columns are permutation invariant.

We can define permutation invariance by some appropriate means. We can use the output of some matrix permuting method, such as the bond energy algorithm [7] or a permuting method related to singular value decomposition [17].

The non-uniqueness of such solutions is not an important issue in this article. However we must justify our approach since it does rely on an array permutation method selected by the user. The resulting non-unique solution is acceptable because our ultimate goals are related to data visualization and exploratory data analysis. Our problem-solving approach is *unsupervised* rather than *supervised*, to use terms which are central in pattern recognition. We seek *an* interpretation of our data, rather than *the* unique interpretation. Of course, the unsupervised data analysis may well precede or be otherwise very closely coupled to supervised analysis (discriminant analysis, statistical estimation, exact database match, etc.) in practice.

From a 2-way data array, a 2-dimensional image is created by considering a point at (x, y) as defining the value 1 at that point, yielding the tuple $(x, y, 1)$; projecting onto a regular discrete grid in the plane; and assigning the contribution of points to the image pixels by means of the interpolation function, ϕ, related to the chosen wavelet transform algorithm – in our case, a B_3 spline.

In a wavelet transform [18, 19], a series of transformations of an image is generated, providing a resolution-related set of “views” of the image. The properties satisfied by a wavelet transform, and in particular the à trous wavelet transform (with holes, so called because of the interlaced convolution used in successive levels: see step 2 of the algorithm below) are further discussed in [20].

The wavelet transform we use is the *à trous* (“with holes”) method (see e.g. [21]). It is a redundant (i.e. non-pyramidal) method and has computational cost which is linear as a function of the number of pixels in the input data. A summary of the à trous wavelet transform is as follows. Index k ranges over all pixels.

1. Initialize i to 0, starting with an image $c_i(k)$.

2. Increment i, and carry out a discrete convolution of the data yielding $c_{i-1}(k)$ using a filter h (see below). The distance between a central pixel and adjacent ones is 2^{i-1}.

3. From this smoothing we obtain the discrete wavelet transform, $w_i(k) = c_{i-1}(k) - c_i(k)$.

4. If i is less than the number p of resolution levels wanted, return to step 2.

The set $\mathcal{W} = \{w_0, w_1, ..., w_p, c_p\}$, where c_p is a residual, represents the wavelet transform of the data. The discrete filter h, when successively applied, realises convolutions with the increasingly dilated B_3 spline function. Implementation is carried out by taking the image dimensions as separable and using the following: $h = \{1/16, 3/8, 1/4, 3/8, 1/16\}$.

The following additive decomposition of the input image follows directly from the above algorithm:

$$c_0(k) = c_p + \sum_{i=1}^{p} w_i(k) \tag{1}$$

Noise filtering of all wavelet resolution scales, followed by reconstitution of the (now filtered) input data, is very effective in practice. It is premised on the following considerations. Linkage or incidence or co-occurrence data of the sort considered here generally contains noise. Hence the wavelet coefficients are noisy too. Therefore we ask ourselves if a wavelet coefficient is due to signal (i.e. it is significant) or to noise. Many noise models can be considered, for different types of data (e.g. multiplicative noise in the case of sonar or radar data, additive noise containing Gaussian and low order Poisson components for widely-used CCDs, charge coupled device detectors, etc.). In the examples discussed later in this paper we consider the additive, stationary, low-count Poisson noise case, which corresponds to random shot noise.

A probability distribution function summarizes all possible eventualities, from the extreme of a uniform arrangement of counts, through to the extreme of all counts being stacked in one pixel. Based on this, a set of signal detection thresholds can be built up for each wavelet resolution level. Murtagh and Starck [2] can be referred to for further details, including the distribution functions for a wide range of numbers of counts.

Having the distribution of the wavelet coefficient for each resolution plane, based on the noise, we can introduce a statistical significance test for this coefficient. This procedure is the classical significance-testing one, where we test the null hypothesis that the image is locally constant at the given resolution scale.

The multiresolution support (Starck et al., 1995) is the name we use for the data structure resulting from noise filtering. It is based on the detection at each scale of the significant wavelet coefficients. The multiresolution support is defined by:

$$M(j, x, y) = \begin{cases} 1 & \text{if } w_j(x, y) \text{ is significant} \\ 0 & \text{if } w_j(x, y) \text{ is not significant} \end{cases} \tag{2}$$

We will say that a multiresolution support of an image describes in a logical or boolean way if an image I contains information at a given scale j and at a given position (x, y). The algorithm to create the multiresolution support is therefore as follows:

1. Compute the wavelet transform of the image.

2. Estimate the noise standard deviation at each scale. Deduce the statistically significant level at each scale.

3. Booleanization of each scale leads to the multiresolution support.

4. Modification using a priori knowledge is carried out if desired.

Note that step 4 allows us to incorporate expert knowledge into the data analysis operation. If we know that a cluster of document-term dependency links is not of interest if it contains a very small number of such links, then we can suppress in the support any cluster below a user-specified cardinality. Mathematical morphology may be a useful tool for doing this.

Armed with our wavelet transform data structure, and our noise filtering procedures, we now proceed to look at the use of this methodology in a range of practical cases.

3 Matrix Reordering

Our methodology rests on (i) permuting the rows and columns of an incidence array to some standard form, and (ii) treating the permuted array as an image, analyzed subsequently by a multiscale transform method.

A few comments on the computational aspects of array permuting methods follow [17]. Gathering larger (or nonzero) array elements to the diagonal can be viewed in terms of minimizing the envelope of nonzero values relative to the diagonal. This can be formulated and solved in purely symbolic terms by reordering vertices in a suitable graph representation of the matrix. A widely-used method for symmetric sparse matrices is the Reverse Cuthill-McKee (RCM) method.

The complexity of the RCM method for ordering rows or columns is proportional to the product of the maximum degree of any vertex in the graph represented by the array and the total number of edges (nonzeroes in the matrix). For hypertext matrices with small maximum degree, the method would be extremely fast. The strength of the method is its low time complexity but it does suffer from certain drawbacks. The heuristic for finding the starting vertex is influenced by the initial numbering of vertices and so the quality of the reordering can vary slightly for the same problem for different initial numberings. Next, the overall method does not accommodate dense rows (e.g., a common link used in every document), and if a row has a significantly large number of nonzeroes it might be best to process it separately; i.e., extract the dense rows, reorder the remaining matrix and augment it by the dense rows (or common links) numbered last.

One alternative approach is based on linear algebra, making use of the extremely sparse incidence data which one is usually dealing with. The execution time required by RCM may well require at least two orders of magnitude (i.e., 100 times) less execution time compared to such methods. However such methods, including for example sparse array implementations of correspondence analysis, appear to be more competitive with respect to bandwidth (and envelope) reduction at the increased computational cost.

Elapsed CPU times for a range of arrays are given in [17], and as an indication show performances between 0.025 to 3.18 seconds for permuting a 4000×400 array.

4 A Case-Study: Fisher's Iris Data

The iris data of Anderson used by Fisher [23] is a very widely-used benchmark dataset. The data consists of 3 varieties of iris flower, each providing 50 samples. There are measurements on 4 variables, petal and sepal length and breadth. The data matrix is therefore one of dimensions 150×4.

To simulate a higher-dimensional problem, and to remain with the familiar properties of the Fisher iris data, we generated a recoded version of the Fisher data. For each of the 150 samples, we took the 4 given measurements, and made a much larger vector from them – as it happens a 147-valued binary (0 or 1) vector. This was done simply by taking each 0.1 interval of each variable as defining a new variable. You can check the veracity of this data recoding by looking at a principal components analysis of the original data, i.e. the optimal 2-dimensional projection of the 4-dimensional space (Fig. 1); and a correspondence analysis of the recoded data, i.e. an optimal 2-dimensional projection of the 147-dimensional space (Fig. 2).

While traditional methods have increased computational difficulty in dealing with higher-dimensional spaces, this is just what a wavelet transform approach thrives on.

We will look at the analysis of the 150 × 147 Fisher data. First we permute rows and columns since we will look for contiguously-formed clusters, and therefore we need a canonical ordering of some sort to make this possible. Methods which give permuted results with the contiguity-enforcing property were overviewed in Sections 1 and 3 above. The permuting method used by us was to take a principal components analysis of the 150 × 147 data and use the order of the principal component projections. (We did not use the correspondence analysis result, even though this is more appropriate for the boolean data used as input, for a minor technical reason: the correspondence analysis deleted columns among the 147 which had zero totals, thus not providing a convenient 147-valued set of ranks for us to work on). The reordered data used is shown in Fig. 3.

The à trous wavelet transform with 5 resolution scales and a residual gives the result shown in Fig. 4. When added, these give a smoothed (with the scaling function used by the à trous method) version of the input data. We suppose now that we are dealing with signal in the form of clusters or clumps of samples/measurements, and further that this nice view of our data is sullied by spurious information in the form of (low-count) Poisson noise, and we filter our data.

The multiscale significant detections based on these assumptions are shown in the sequence of images in Fig. 5.

The 5th scale, here, looks the most informative in terms of our set of 150 samples, divided as we know into three classes each of 150. The samples are represented on the horizontal axis. Taking the positive parts of this image, and taking everything above 0 as equal to 1 gives the visualization shown in Fig. 6.

Reading off the sample numbers from the horizontal axis gave a credible result. For the first Fisher cluster, corresponding to the upper right cluster, one substitution error was found. The other Fisher clusters are a little less resolved, but – cf. the correspondence analysis output of the 150 × 147 data shown in Fig. 2 – are quite consistent with the input.

In dealing with the curse of dimensionality, therefore, we have a method which is fast, provides a good-quality result, and is not hampered by high dimensionality.

5 Ultrametric Spaces

It is interesting to look at the wavelet transform of spaces of known structure. Ultrametric distance matrices can be represented, subject to an appropriate ordering of objects, with quite particular relations between values as we move away from the diagonal. Lerman [24] discusses ultrametric spaces in detail. Lerman's Theorem 2 (1981, p. 45) describes properties of ultrametric distance matrices. The result we are most interested in is in regard to matrix reordering: an order can be found such that array elements are necessarily non-increasing as we move away from the diagonal, and row and column array elements have a number of such inequality properties. We will visualize these properties using a wavelet transform.

To derive ultrametric distances, we again took the Fisher iris data, in its original 150×4 form. We constructed a complete link hierarchical clustering, using the Euclidean distance between the observation vectors. We read off the 150×150 ultrametric distances (ranks were used, rather than agglomeration criterion values) from this dendrogram. Fig. 7 (left) shows this ultrametric matrix. (The greyscale values have been histogram-equalized for better contrast.) When we reorder the rows and columns (the matrix is symmetric of course) in accordance with the ordering of singletons used by the dendrogram representation we get the visualization shown in Fig. 7 (right). Again contrast stretching through histogram-equalization was used.

Figures 8 and 9 show, respectively, the wavelet transforms of the arrays shown in Fig. 7. Three wavelet resolution levels have been used, together with the smooth data residual. They are read from top left to bottom right. Fig. 9 shows a visual representation of Lerman's Theorem 2. Note that the unpermuted data (Fig. 7 left, and Fig. 8) would not usually, in practice, have an order consistent with near separation of the more important clusters, as is the case here.

6 Clustering of Document-Term Data

Experiments were carried out on a set of bibliographical data – documents in the literature crossed by user-assigned index terms. This bibliographic data is from the journal *Astronomy and Astrophysics* (Springer-Verlag). It is used currently to provide a cluster-based graphical user interface to further information on these articles, and in many cases (if one's library subscribes to the journal) to the full online articles themselves. This document map can be accessed at URL http://cdsweb.u-strasbg.fr/Abstract.html. Further information on the construction and maintenance of these document maps is available in [25, 26]. We looked at a set of such bibliography relating to 6885 articles published in *Astronomy and Astrophysics* between 1994 and early 1999. A sample of the first 10 records is as follows.

```
1994A&A...284L...1I 102 167
1994A&A...284L...5W 4 5 14 16 52 69
1994A&A...284L...9M 29
1994A&A...284L..16F 15 64
1994A&A...284....1B 32 49 71
1994A&A...284...12A 36 153 202
1994A&A...284...17H 3 10 74 82 103
1994A&A...284...28M 17 42 102
1994A&A...284...33D 58
1994A&A...284...44S 111
```

A 19-character unique identifier (the *bibcode*) is followed by the sequence numbers of the index terms. There are 269 of the latter. They are specified by the author(s) and examples will be seen below towards the end of this section. The experiments to follow were based on the first 512 documents in order to facilitate presentation of results. Fig. 10 shows the 512×269 incidence array used. We investigated the row and column permuting of this array, based on the ordering of projections on the principal component, but limited clustering was brought about. This was due to the paucity of index term "overlap" properties in this dataset, i.e. the relatively limited numbers of index terms shared by any given pair of documents. For this reason, we elected to base subsequent work on the contingency table. Fig. 11 shows this. Actual values between pixels (document sequence numbers) 251 and 265 are shown as follows:

```
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  4  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  2  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  3  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  2  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  2  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  6  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  3  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  5  0  0  0  0  0
0  0  0  0  0  0  1  0  0  0  4  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  3  0  1
0  0  0  0  0  0  0  0  0  0  0  0  0  3  0
0  0  0  0  0  0  0  0  0  0  0  0  1  0  4
```

A principal components analysis of the 512×269 dataset is dominated by the $O(m^3)$, $m = 269$ diagonalization requirement. Calculating the principal component projections for the rows takes linear (in document space) time. We used the order of principal component projections to provide a standard permutation of rows and columns of the document contingency table. The resulting permuted contingency table is shown in Fig. 12. Actual values between pixels (document sequence numbers) 251 and 265 are as follows:

```
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  3  0  0  1  0  0  0  0  0  0  0  0  0
0  0  0  2  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  2  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  3  0  0  0  0  0  0  0  0  1
0  0  0  0  0  0  2  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  2  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  3  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  4  0  1  0
0  0  0  0  0  0  0  0  1  0  0  0  2  0  0
0  0  0  0  0  0  0  0  0  0  0  1  0  3  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  5
```

A first investigation was carried out on denoising of the permuted contingency table data (i.e. Fig. 12). Clearly though, there is "porosity" here, such that the areas sought which are significantly above the background are not fully contiguous. Denoising of Fig. 12 yielded a view of the data which was almost entirely cleaned in non-diagonal areas. The diagonal itself is given by the first principal component. We looked at the major cluster "knots" in the denoised image but found it of limited usefulness to interpret them. We therefore took a different tack.

Figs. 13 and 14 show, respectively, the results of a wavelet transform (the redundant à trous transform is used) at wavelet resolution level 3 and the final smoothed version of the data. The latter is a background or continuum. In both of these figures, more especially in the continuum one, we have visual evidence for a cluster at the bottom left, another smaller one about one-third of the way up the diagonal, and a large one centred on the upper right-hand side of the image.

We are simply using the wavelet transform in this instance to facilitate analysis of a large, permuted data array. We wish to find contiguous clusters. Such clusters will for the most part be close to the diagonal. We recall that the contingency array used is symmetric, which explains the symmetry relative to the diagonal in what we see.

We can interpret the clusters on the basis of their most highly associated index terms. This in turn relates to the ordering of index terms on the first principal component axis in this case. Applying an arbitrary cut-off (±0.2) to principal component projections, we find the index terms most associated with the two ends of the first principal component as follows:

```
stars:circumstellar matter
X-rays:stars
stars:abundances
stars:evolution
stars:mass loss
stars:binaries:close
stars:late type
stars:activity
stars:magnetic fields
stars:coronae
stars:flare
radio continuum:stars
stars:chromospheres
stars:binaries
```

The other extremity of the first principal component axis is associated with the following index terms:

```
ISM:molecules
galaxies:ISM
galaxies:kinematics and dynamics
galaxies:evolution
galaxies:spiral
galaxies:interactions
galaxies:structure
galaxies:abundances
galaxies:redshifts
galaxies:luminosity function,mass function
galaxies:compact
```

The distinction is clear – between stars, and stellar topics of inquiry, on the one hand, and interstellar matter (ISM) and galaxies, i.e. topics in cosmology, on the other hand. This distinction explains the two clusters clearly visible at the opposite ends of the diagonal in Fig. 14 (and less so in the original permuted data, Fig. 12). The distinction between stellar and cosmological foci of inquiry in the astronomical literature is a well-known one, which is linked directly and indirectly to astronomical instrumentation and even to shifts of interests of professional astronomers over the past few decades.

7 Application to Hypertext

From the Concise Columbia Encyclopedia (1989 2nd ed., online version) a set of data relating to 12025 encyclopedia entries and to 9778 cross-references or links was used. This data was rebinned 10-fold for computational convenience to produce the 1203 × 978 entries × links array used here. The image in Fig. 15 shows part of this array, subsequent to row/column permutation to force nonzero values as far as possible onto the diagonal. The permutation method used was a sparse matrix method related to correspondence analysis diagonalization (Berry et al., 1996).

Now we seek to find the most important "knots" or clusters. Fig. 16 is a filtered version, i.e. with the effects of the noise removed on all resolution scales. We have achieved a very great economy of information, pinpointing salient clusters which, in conjunction with the input data, can be used for search navigation or for further more detailed analysis (for example, causal network modeling).

In Figs. 15 and 16, the first coordinate runs from left to right, and the second from top to bottom. We will explore one cluster found, approximately in the center of Figs. 15 and 16. With reference to the latter Fig., there is a three-some in the center, and we take the leftmost of these isolated clusters. This leftmost cluster is of dimensions 2 × 2. This cluster, and the original data, in this region looks like the following (respectively, first and second of these data "chunks"):

```
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   2   2   0   0   0   0
0   0   0   0   2   2   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
```

```
1   0   0   0   1   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   1   0   1   0   0   0
0   2   0   1   0   0   0   0   0   0
0   0   1   2   1   0   0   0   1   0
```

```
 0   0   0   0   2  10   3   0   0   1
 0   0   0   0   0   0   0   3   0   0
 0   0   0   0   0   0   0   0   4   1
 0   0   0   0   1   0   0   0   1   0
 0   0   0   1   0   0   0   0   0   0
```

The larger region with the other clusters in the three-some is as follows (again, filtered data, and original data, respectively):

```
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   2   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   2   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   1   2   1   1   1   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1   3   1
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   3   1
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 2   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0
 0   2  10   3   0   0   1   0   0   1   0   0   0   1   0   0   0   0   0
 0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   4   1   0   0   1   0   0   0   0   0   0   0   0   0
 0   1   0   0   0   1   0   3   5   2   0   0   0   0   0   0   0   0   0
 1   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0
 0   0   0   0   1   0   0   0   0   0   2   0   1   0   0   1   0   0   0
 0   0   0   0   0   0   0   0   0   0   1   2   0   1   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   1   0   2   6   0   0   0   0
 0   0   0   0   0   0   0   1   0   1   0   0   0   0   1   5   3   0   0
 1   1   0   0   0   1   0   1   0   0   0   0   0   0   0   0   0   8   9
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0
```

This gives an instructive visual impression of our approach. We see that large values will serve to define clusters. We also see that there is a strong aspect of smoothing, based on our algorithm's understanding that relationships have to be enforced between large valued-pixels to form clusters.

Investigation of clusters is complicated by the fact that we have an incidence array of dimensions 12025 × 9778, which we rebin to dimensions 1203 × 977, because of limitations on available memory. Such rebinned data can be checked out more closely, on the basis of the clustering carried out. We will not pursue further here the information navigation possibilities which ensue.

To look further at a cluster or two, we instead returned to the original data. We took the first 1203 × 977 values, based on the correspondence analysis reordering. About the lower half of this array was very much diagonalized, and was therefore relatively straightforward to analyze. (The clusters in fact formed a one-dimensional ordering or seriation, and therefore were particularly easy to process.) The upper part of the array was more dispersed and this is what we analyzed using our method. Fig. 17 shows this 500 × 450 array.

This part of the encyclopedia data was filtered using the wavelet and noise-modeling methodology described in this paper. Since we are interested in crisp clusters here, this filtered result was thresholded and a boolean image was created from this. Then, to remove the influence of what had been blurred clusters, which are not of interpretative value to us since again the relations in our data are non-fuzzy, we took the product (intersection) of the original incidence data and the processed filtered data. The result is shown in Fig. 18.

Overall the recovery of the more apparent alignments, and hence visually stronger clusters, is excellent. The first relatively long "horizontal bar" was selected – it corresponds to column index (link) 1733 = `geological era`. The corresponding row indices (articles) are, in sequence:

```
SILURIAN PERIOD
PLEISTOCENE EPOCH
HOLOCENE EPOCH
PRECAMBRIAN TIME
CARBONIFEROUS PERIOD
OLIGOCENE EPOCH
ORDOVICIAN PERIOD
TRIASSIC PERIOD
CENOZOIC ERA
PALEOCENE EPOCH
MIOCENE EPOCH
DEVONIAN PERIOD
PALEOZOIC ERA
JURASSIC PERIOD
MESOZOIC ERA
CAMBRIAN PERIOD
PLIOCENE EPOCH
CRETACEOUS PERIOD
```

One remark to be made is that the filtering program was asked to ignore any clusters touching the array boundaries due to possible difficulties with interpretation – cf. lower left alignments – and this is something which can be changed easily. We can see some preference for vertical and horizontal alignments, over diagonal ones, which is reasonable – the former represent clusters of encyclopedia entries and of cross-references. There is scope in our method for fine-tuning this objective, for example to prioritize clustering of entries over those of cross-references.

8 Conclusion

This paper describes a new set of linkages between wavelet transform analysis and multivariate data analysis (see [27] for various other linkages, especially to dimensionality reduction methods). Exhibits of wavelet transforms, for applications related to pattern recognition, data enhancement, filtering, compression, data fusion, information characterization, and many others, using a methodology similar to that used here, can be found at MR/1 [28].

The methodology developed here is fast and effective. It is based on the convergence of a number of technologies: (i) data visualization techniques; (ii) the wavelet transform for data analysis; and (iii) data matrix permuting techniques. We have discussed its use for large incidence arrays. We introduced noise modeling of such data, and showed how noise filtering can be used to provide as output a set of significant clusters in the data. Such clusters may be overlapping. Further development of this work would be to investigate hierarchical clusters, possibly overlapping, derived from the multiple scales.

We have also discussed this innovative methodology using a range of different datasets. It is clearly related to other well-established data analysis methods, such as seriation (one-dimensional ordering of observations), and nonparametric density estimation (the wavelet transform can be viewed as performing such density estimation).

We can note also the potential use of our new methodology for use in graphical user interfaces. The Kohonen self-organizing feature map, by now quite widely used for support of clickable user interfaces [25, 26], presents a map of the documents (say), but not as explicitly of the associated index terms. Our maps cater equally for both documents and index terms. Furthermore, the way is open to the exploration of what can be offered by recent developments in client-server based image storage and delivery (see some discussion in Chapter 7 of [3]) e.g. progressive transmission and foveation (i.e. progressive transmission in a local region) strategies. This perspective opens up onto a line of inquiry which could be characterized as *multiple resolution information storage, access and retrieval.*

Our approach responds well to current requirements for fast methods to process large incidence arrays. Its main practical performance limitation is related to machine memory storage for the arrays being produced and analyzed. Such a limitation can be bypassed by processing segments of the incidence array in sequence (or in parallel). The array permuting strategy used will be of major help in this regard also, since it focuses our attention on particular – usually near-diagonal – parts of the reordered incidence array.

Acknowledgements

The suggestion to look at the wavelet transform of ultrametric data was made by A.D. Gordon, St. Andrews. This article also benefited from discussions with participants following keynote presentations at the Japanese Classification Society Annual Meeting (Tokyo, March 1999), and at the Annual Conference of the Classification Society of North America (Pittsburgh, June 1999).

References

[1] Bellman, R. (1961) *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton.

[2] Murtagh, F. and Starck, J.L. (1998) Pattern clustering based on noise modeling in wavelet space, *Pattern Recognition*, **31**, 847–855.

[3] Starck, J.L., Murtagh, F. and Bijaoui, A. (1998) *Image and Data Analysis: The Multiscale Approach.* Cambridge University Press, Cambridge.

[4] Chereul, E., Crézé, M. and Bienaymé, O. (1997) 3D wavelet transform analysis of Hipparcos data, in Maccarone, M.C., Murtagh, F., Kurtz, M. and Bijaoui, A. (eds.). *Advanced Techniques and Methods for Astronomical Information Handling*, Observatoire de la Côte d'Azur, Nice, France, 41–48.

[5] Banfield, J.D. and Raftery, A.E. (1993) Model-based Gaussian and non-Gaussian clustering, *Biometrics*, **49**, 803–821.

[6] Strang, G. and Nguyen, T. (1996) *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley.

[7] McCormick, W.T., Schweitzer, P.J. and White, T.J. (1972) Problem decomposition and data reorganization by a clustering technique, *Operations Research*, **20**, 993–1009.

[8] Lenstra, J.K. (1974) Clustering a data array and the traveling-salesman problem, *Operations Research*, **22**, 413–414.

[9] Doyle, J. (1988) Classification by ordering a (sparse) matrix: a simulated annealing approach, *Applied Mathematical Modelling*, **12**, 86–94.

[10] Deutsch, S.B. and Martin, J.J. (1971) An ordering algorithm for analysis of data arrays, *Operations Research*, **19**, 1350–1362.

[11] Streng, R. (1991) Classification and seriation by iterative reordering of a data matrix, in Bock, H.-H. and Ihm, P. (eds.). *Classification, Data Analysis and Knowledge Organization Models and Methods with Applications*, Springer-Verlag, Berlin, pp. 121–130.

[12] Packer, C.V. (1989) Applying row-column permutation to matrix representations of large citation networks, *Information Processing and Management*, **25**, 307–314.

[13] Murtagh, F. (1985) *Multidimensional Clustering Algorithms.* Physica-Verlag, Würzburg.

[14] March, S.T. (1983) Techniques for structuring database records, *Computing Surveys*, **15**, 45–79.

[15] Arabie, P., Schleutermann, S., Dawes, J. and Hubert, L. (1988) Marketing applications of sequencing and partitioning of nonsymmetric and/or two-mode matrices, in Gaul, W. and Schader, M. (eds.), *Data, Expert Knowledge and Decisions.* Springer-Verlag, Berlin, pp. 215–224.

[16] Gale, N., W.C. Halperin and Costanzo, C.M. (1984) Unclassed matrix shading and optimal ordering in hierarchical cluster analysis, *Journal of Classification*, **1**, 75–92.

[17] Berry, M.W., Hendrickson, B. and Raghavan, P. (1996) Sparse matrix reordering schemes for browsing hypertext, in *Lectures in Applied Mathematics (LAM) Vol. 32: The Mathematics of Numerical Analysis*, Renegar, J., Shub, M. and Smale, S. (eds.). American Mathematical Society, pp. 99–123 (http://www.cs.utk.edu/~berry/order/index.html).

[18] Morlet, J., Arens, G., Fourgeau, E. and Giard, D. (1982) Wave propagation and sampling theory I and II, *Geophysics*, **47**, 203–236.

[19] Mallat, S. (1989) A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 674–693.

[20] Bijaoui, A., Starck, J.L. and Murtagh, F. (1994) Restauration des images multi-échelles par l'algorithme à trous, *Traitement du Signal*, **11**, 229–243.

[21] Shensa, M.J. (1992) The discrete wavelet transform: wedding the à trous and Mallat algorithms, *IEEE Transactions on Signal Processing*, **40**, 2464–2482.

[22] Starck, J.L., Bijaoui, A. and Murtagh, F. (1995) Multiresolution support applied to image filtering and deconvolution, *Graphical Models and Image Processing*, **57**, 420–431.

[23] Fisher, R.A. (1936) The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, **7**, 179–188.

[24] Lerman, I.C. (1981) *Classification et Analyse Ordinale des Données.* Dunod, Paris.

[25] Poinçot, P., Lesteven, S. and Murtagh, F. (1998), A spatial user interface to the astronomical literature, *Astronomy and Astrophysics Supplement Series*, **130** 183–191.

[26] Poinçot, P., Lesteven, S. and Murtagh, F. (1999), Maps of information spaces: assessments from astronomy, submitted to *Journal of the American Society for Information Science.*

[27] Murtagh, F. (1998) Wedding the wavelet transform and multivariate data analysis, *Journal of Classification*, **15**, 161–183.

[28] MR/1 (1999), Multiresolution Image and Signal Analysis Software Environment, Examples of Applications, http://www.multiresolution.com

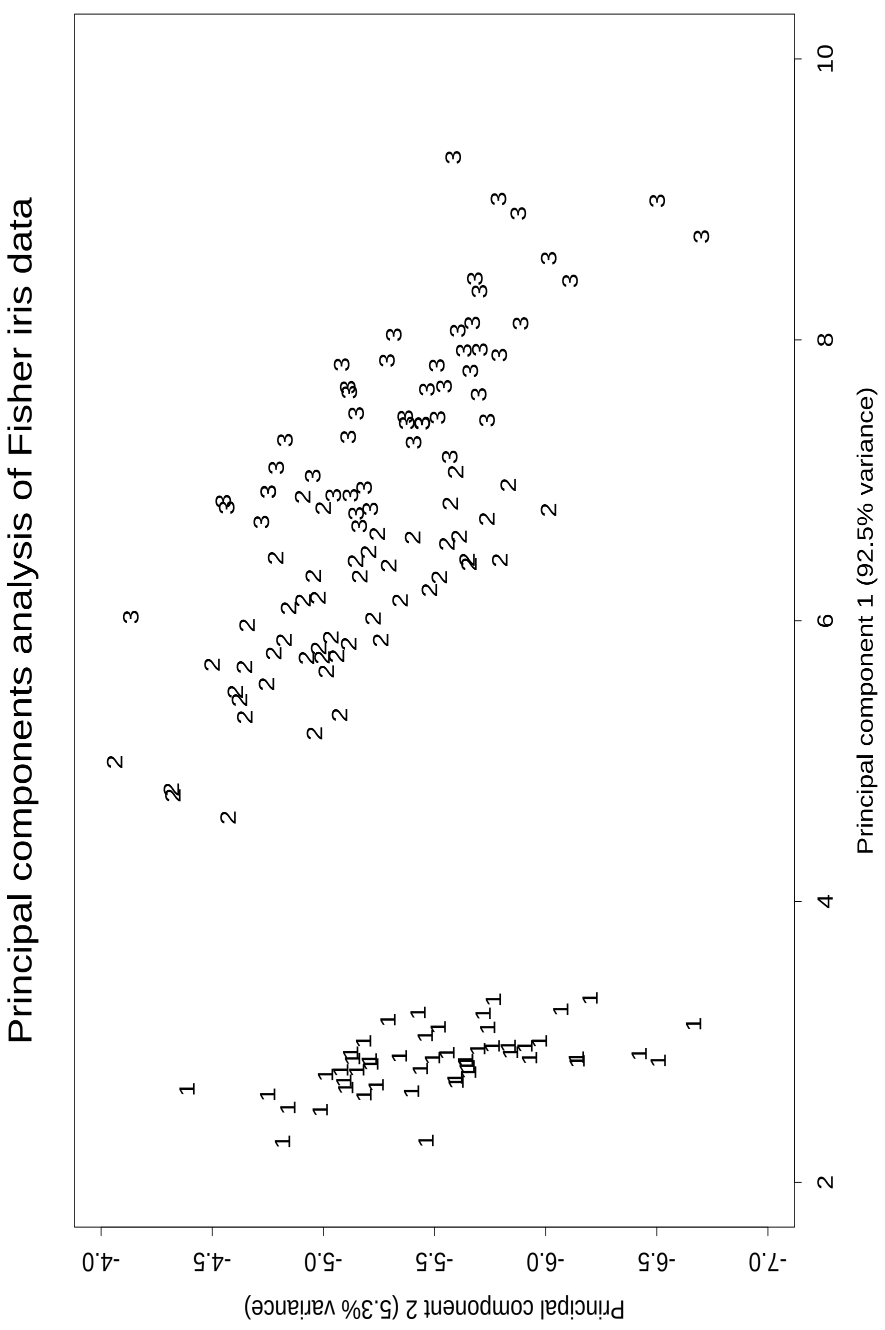

Figure 1: Principal plane of the 4-dimensional Fisher data.

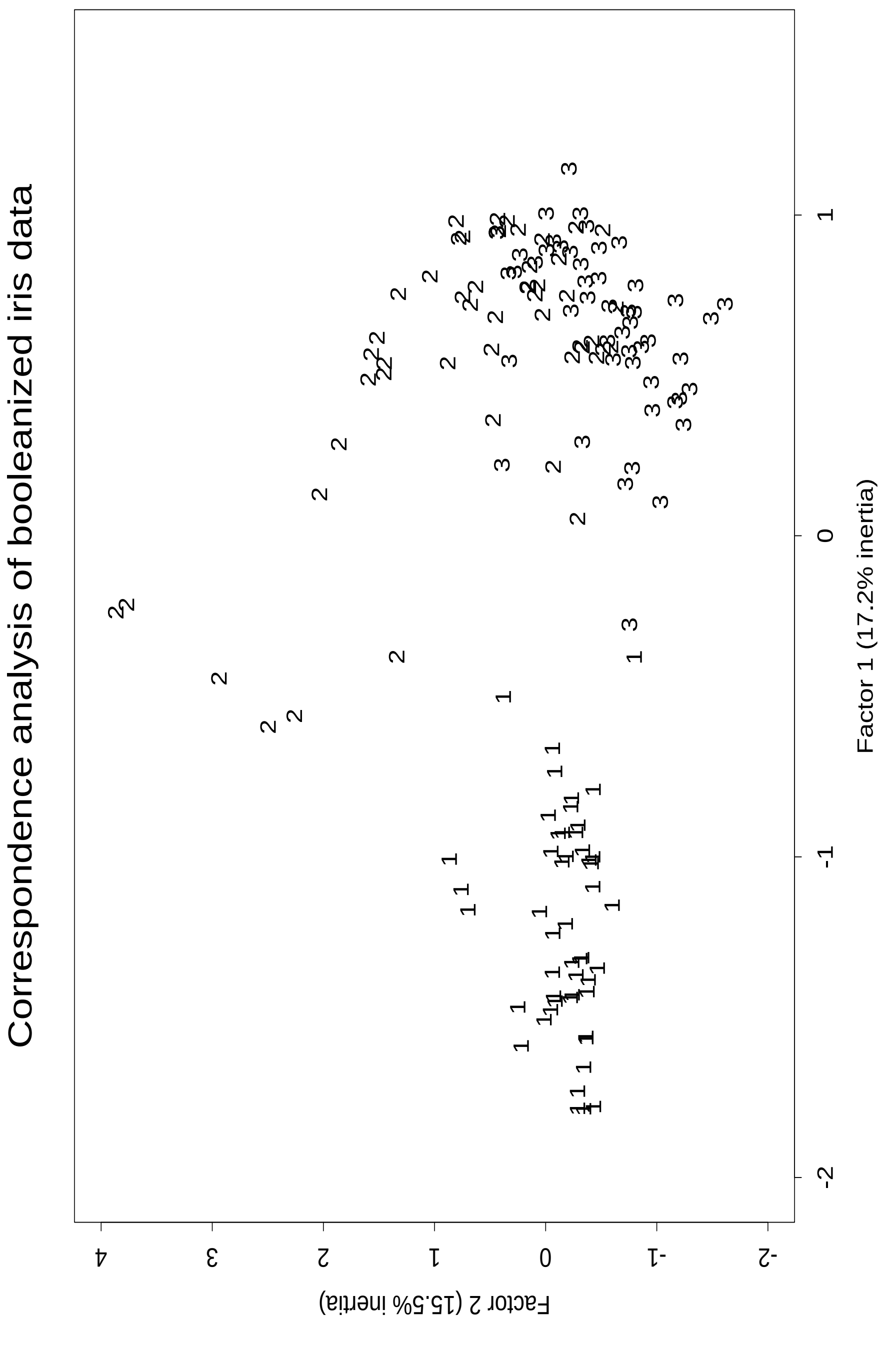

Figure 2: Correspondence analysis, principal factors, of the Fisher data embedded in a high-dimensional (147-dimensional) space.

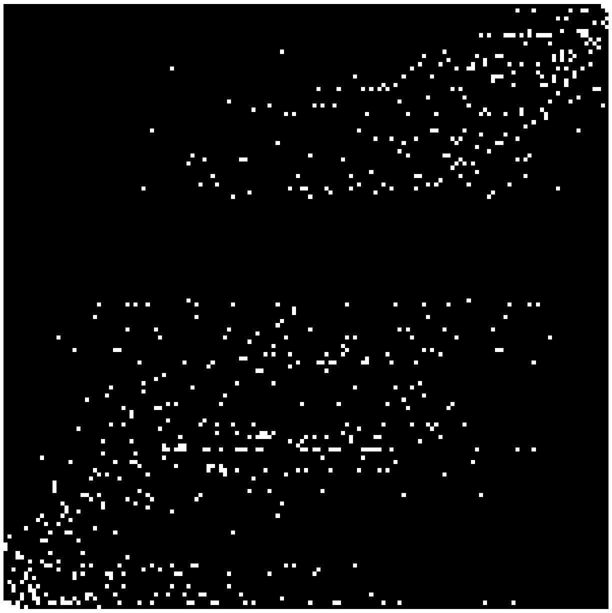

Figure 3: Visualization of reordered version of high-dimensional Fisher data.

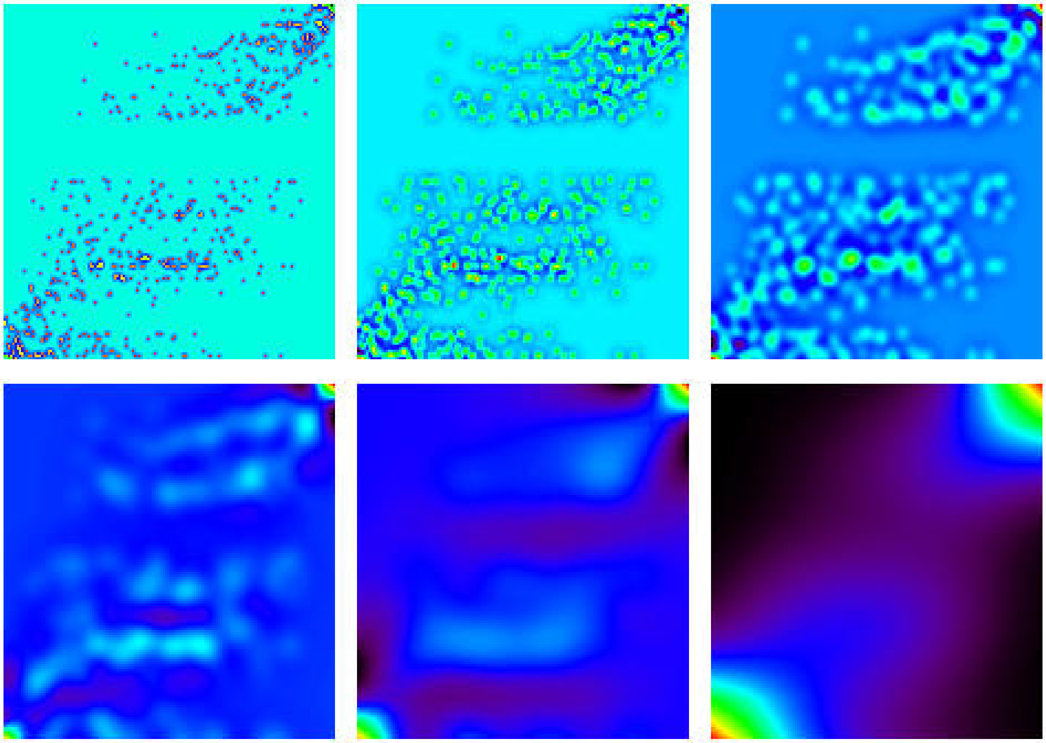

Figure 4: Wavelet transform of reordered Fisher data.

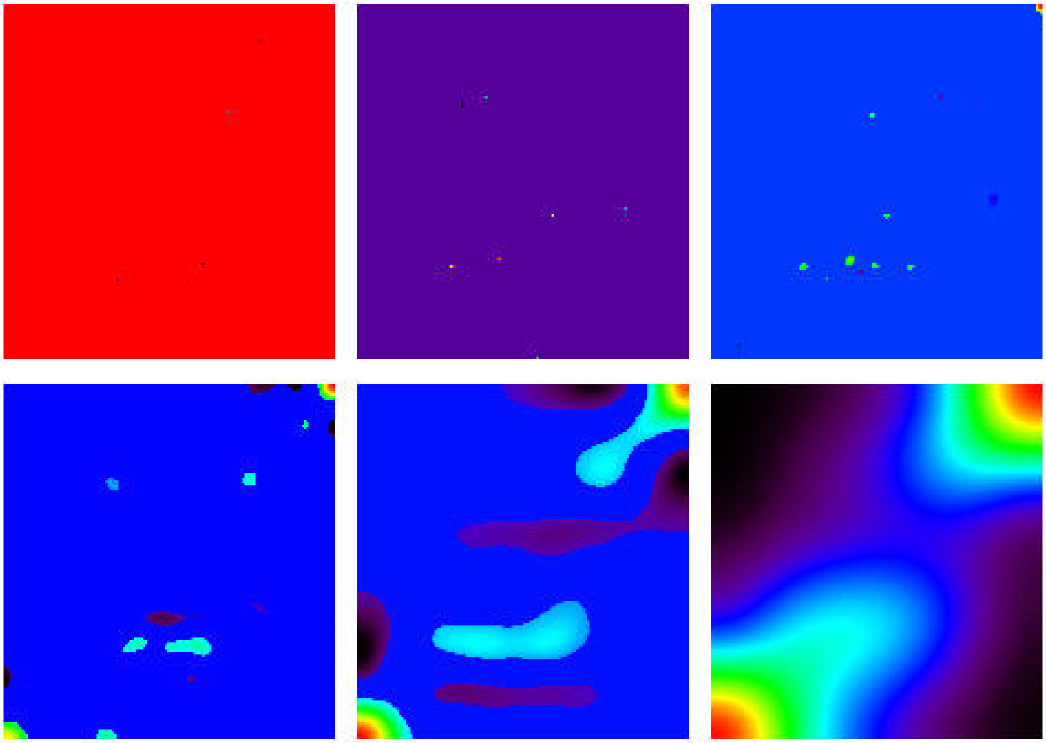

Figure 5: Significant detections in wavelet transform of reordered Fisher data.

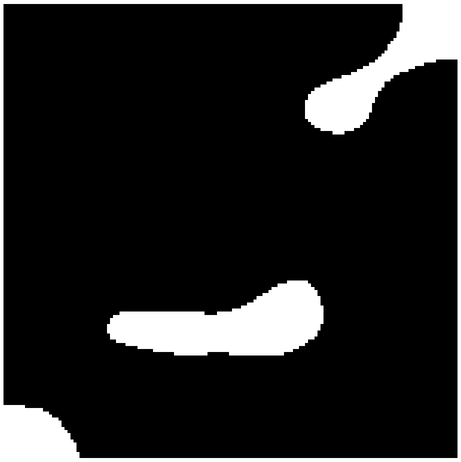

Figure 6: The support image of the 5th scale of the wavelet transform of the Fisher data.

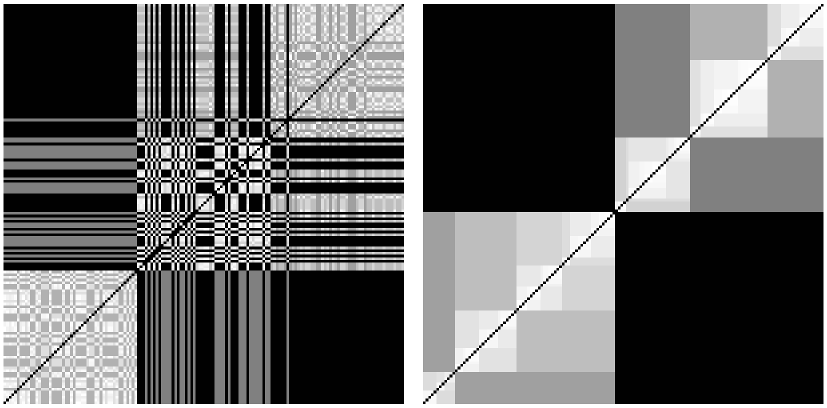

Figure 7: Left: ultrametric matrix of 150 observations, in given order – clusters 1, 2 and 3 correspond to sequence numbers 1–50, 51–100, 101–150. Right: ultrametric matrix of these same observations, with the rows and columns permuted in accordance with a non-crossover representation of the associated dendrogram.

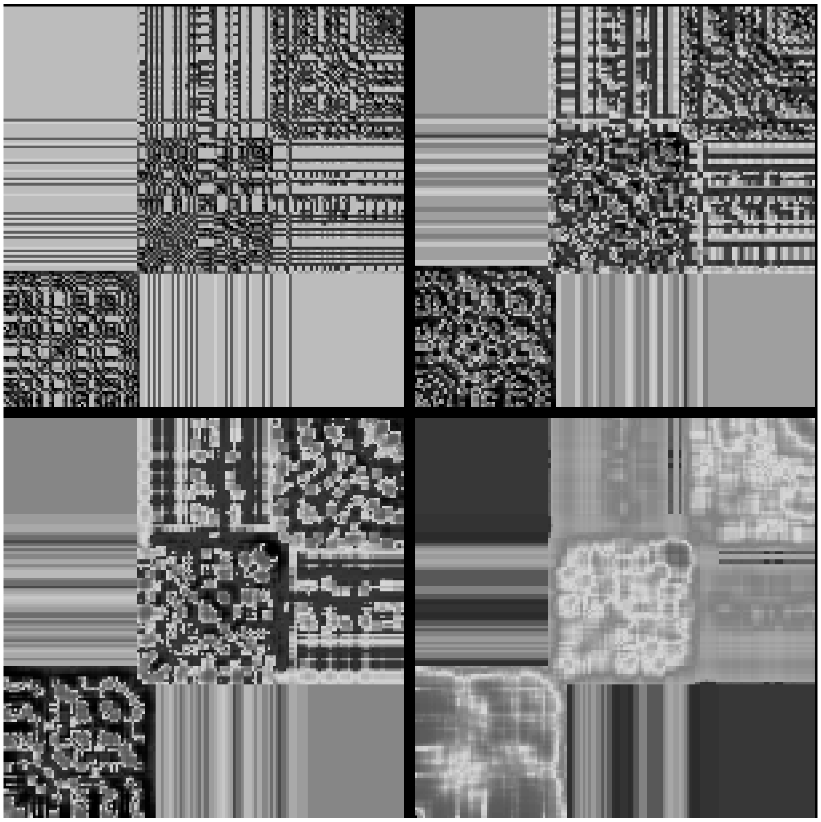

Figure 8: Wavelet transform of array shown in Fig. 7 (left).

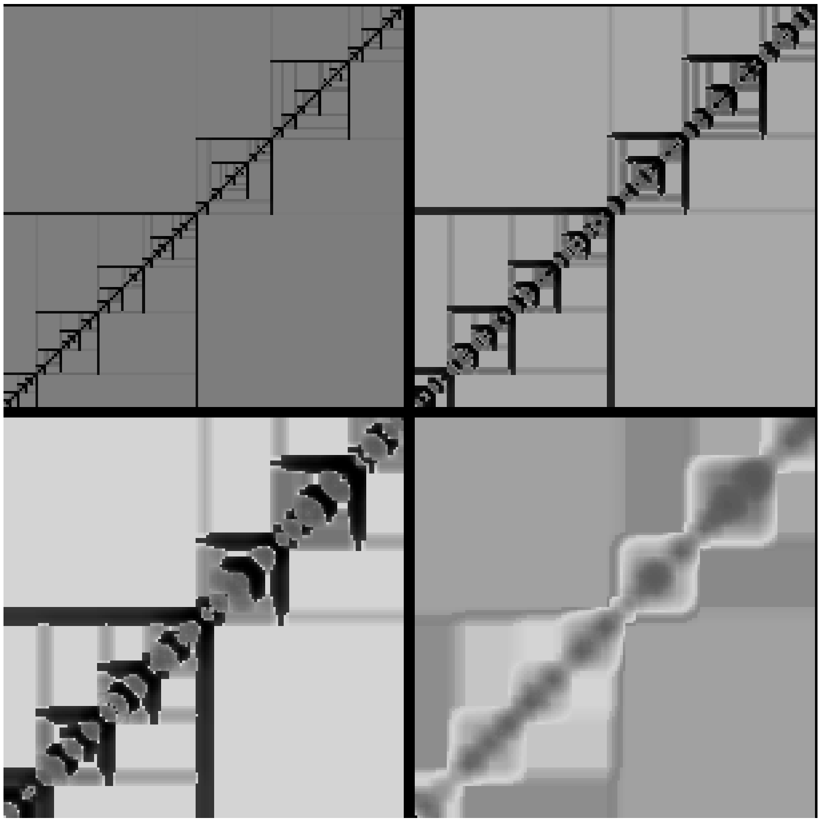

Figure 9: Wavelet transform of array shown in Fig. 7 (right).

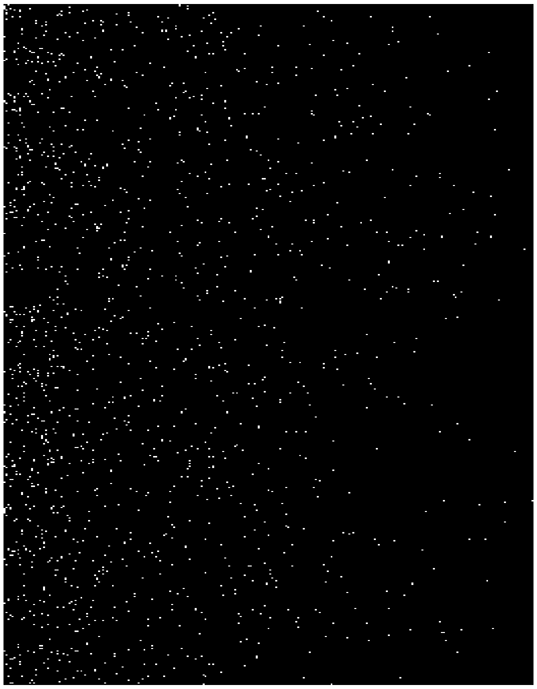

Figure 10: Rows: 512 documents, columns: 269 index terms.

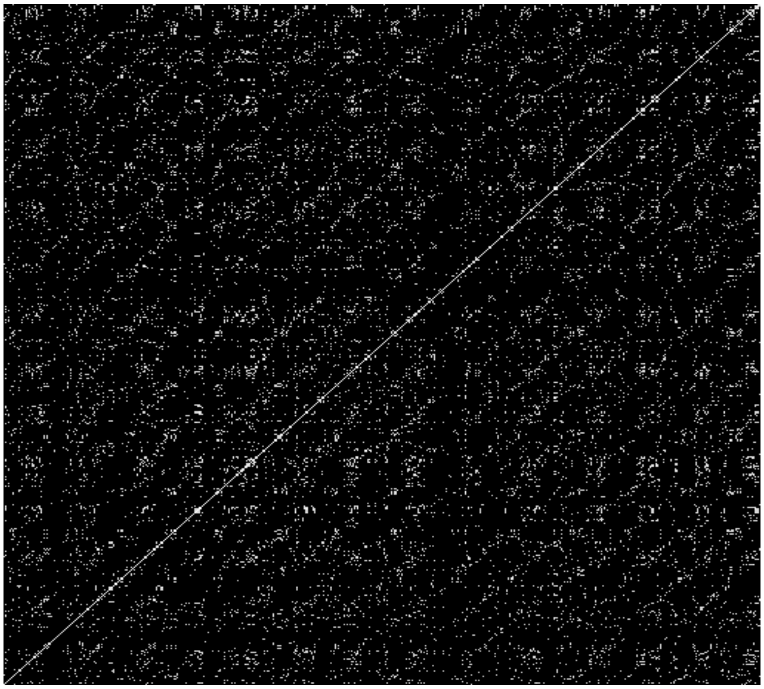

Figure 11: Contingency table of 512 documents.

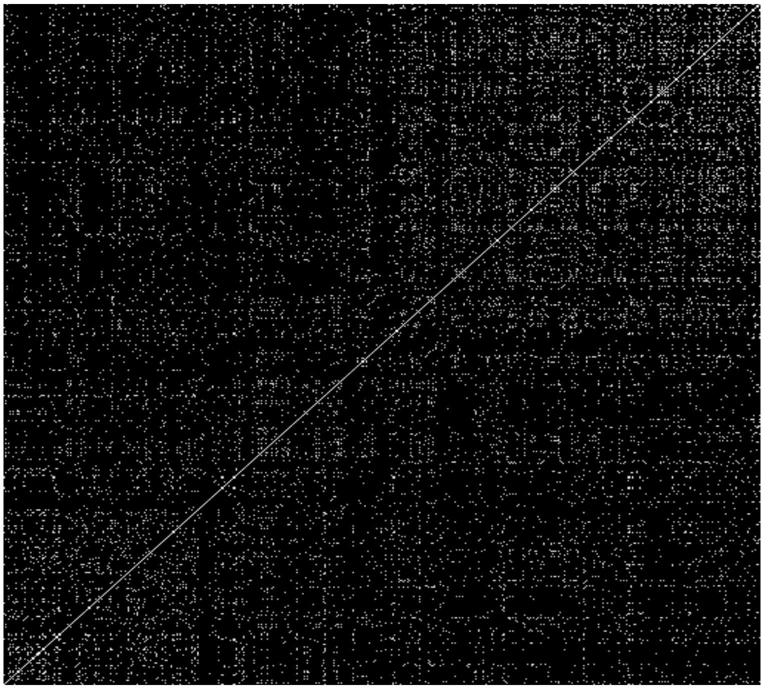

Figure 12: Row/column-permuted contingency table of 512 documents, based on projections onto first principal component.

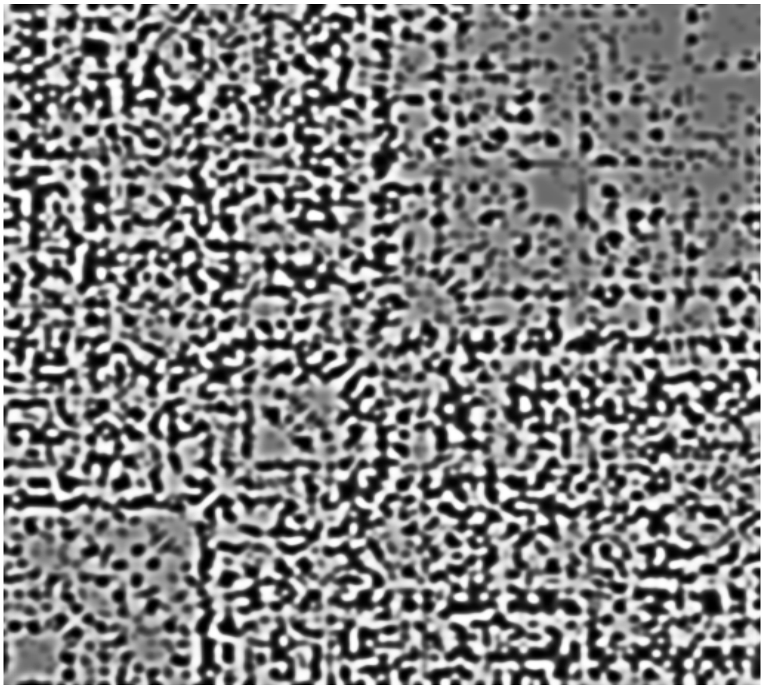

Figure 13: Resolution level 3 from a wavelet transform of the data shown in Fig. 12.

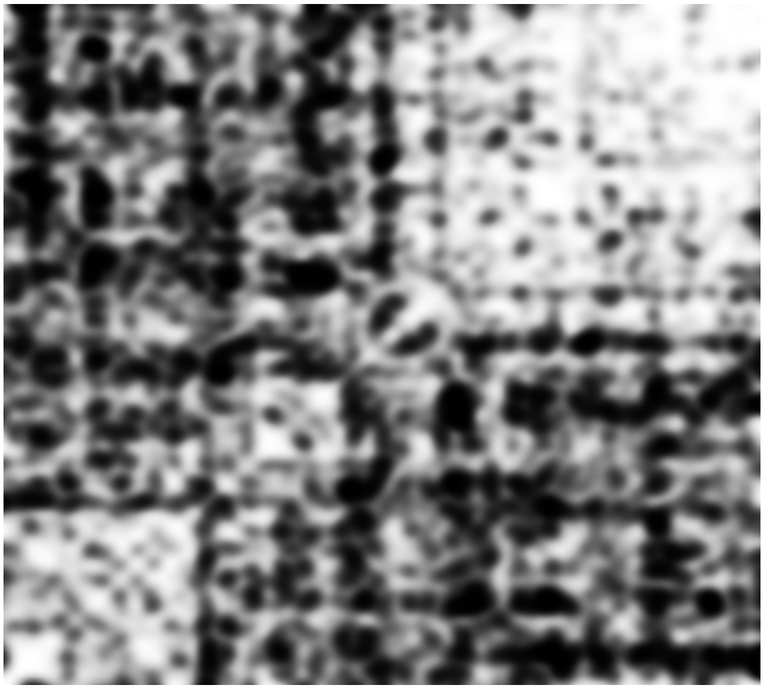

Figure 14: The final smoothed version of the data resulting from a wavelet transform of the data shown in Fig. 12.

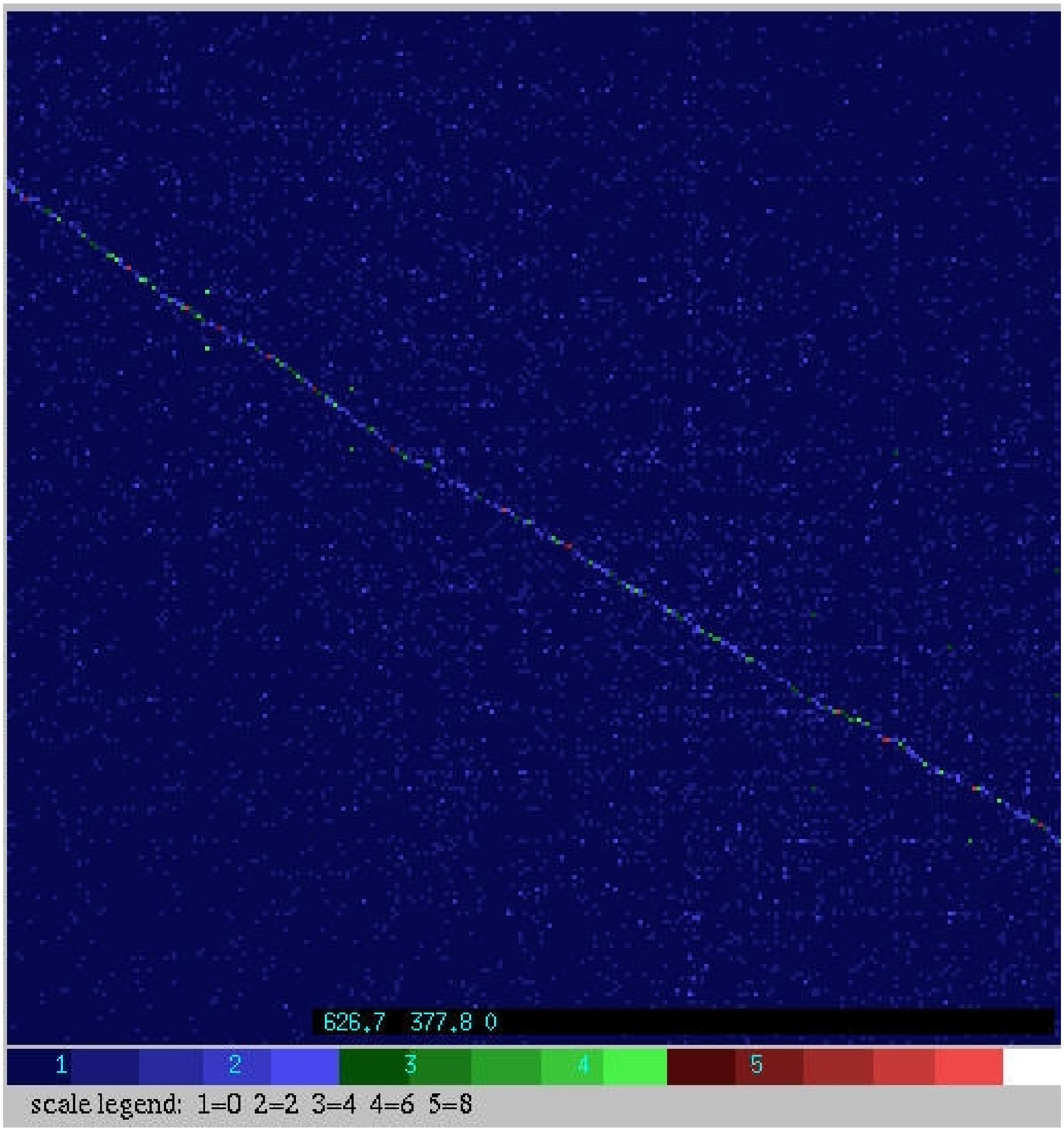

Figure 15: Part of the encyclopedia link dependencies.

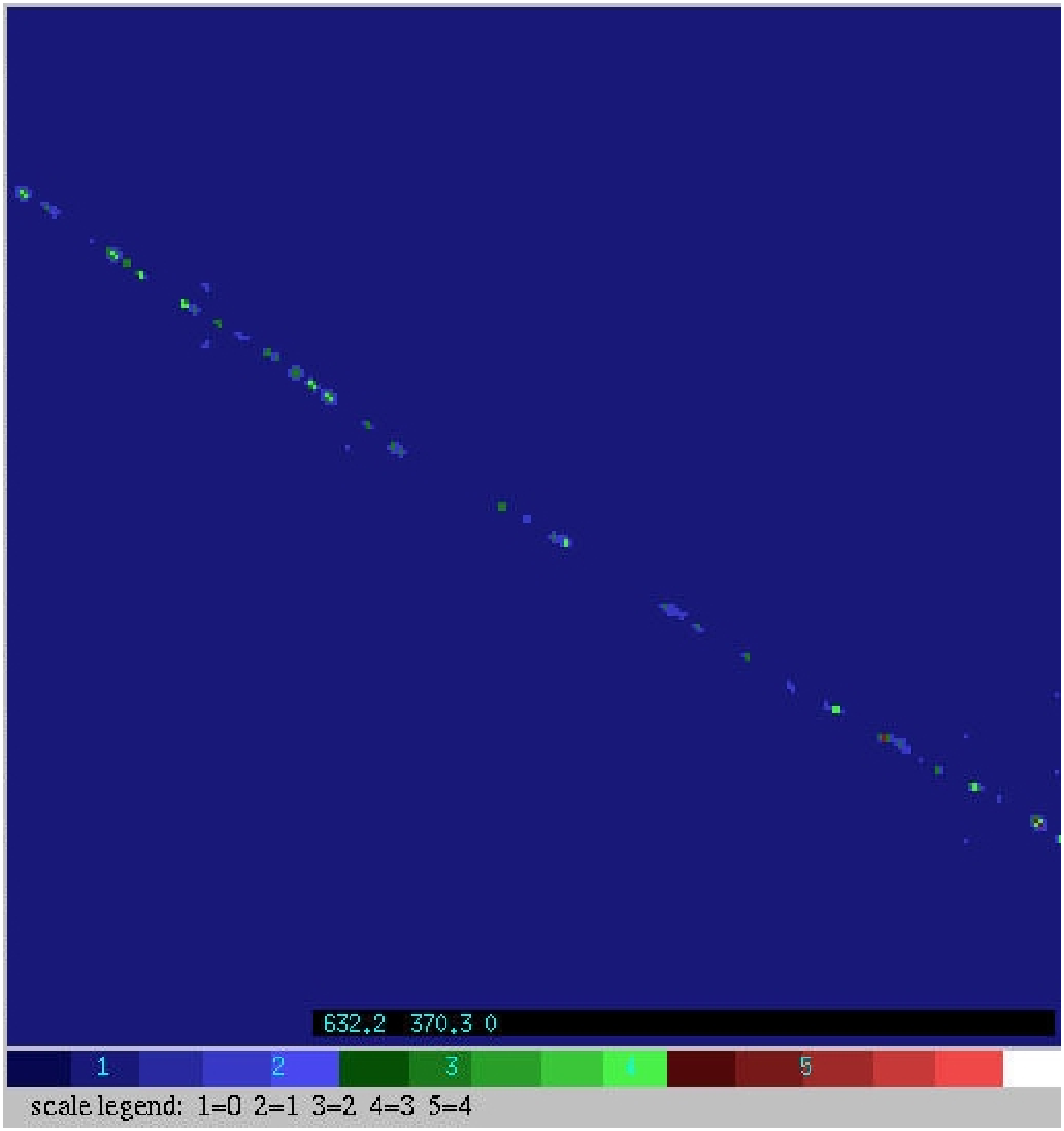

Figure 16: Wavelet filtered version of the data in the approximate region shown in the previous Fig.

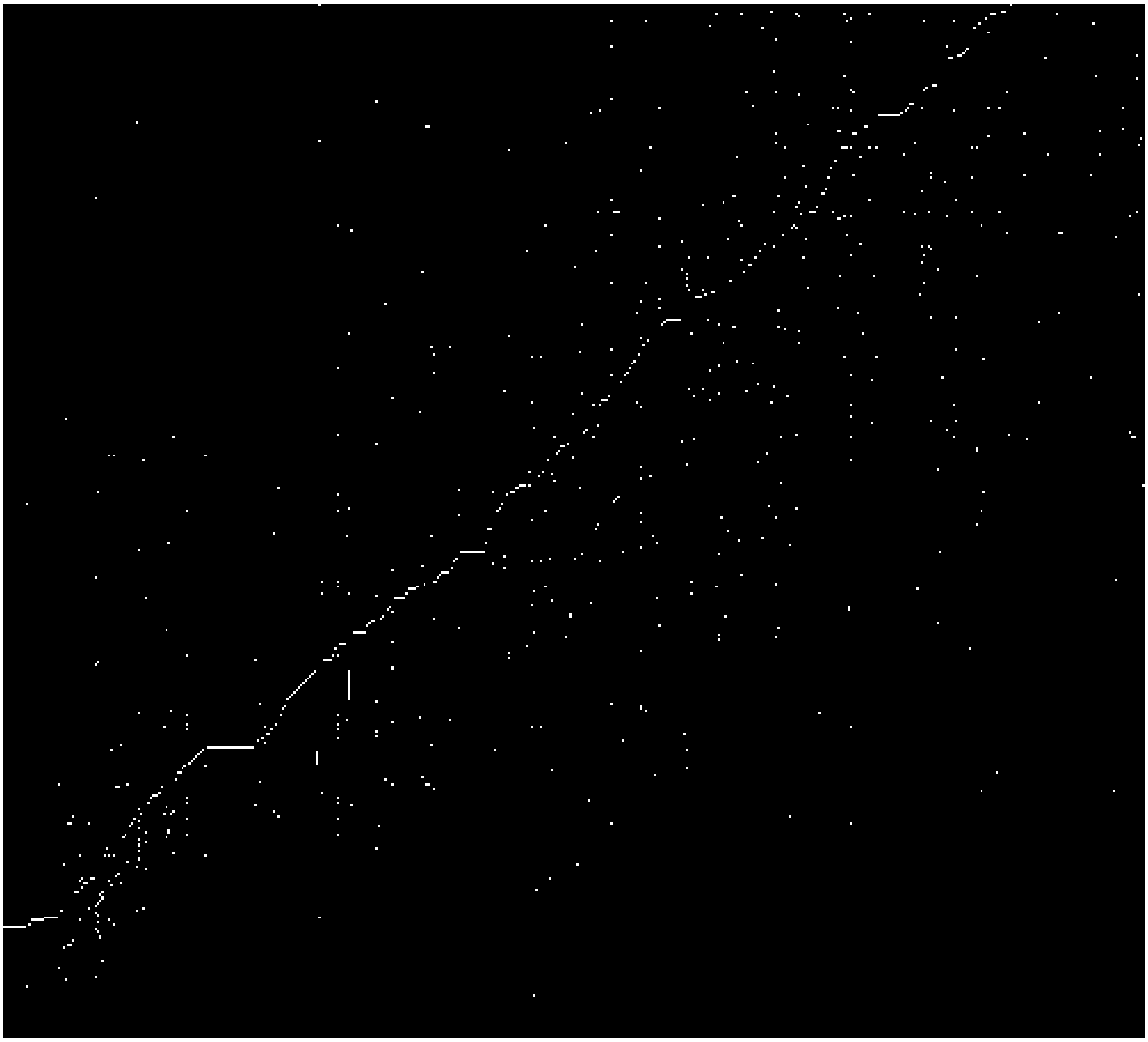

Figure 17: Part (500 × 450) of original encyclopedia incidence data array.

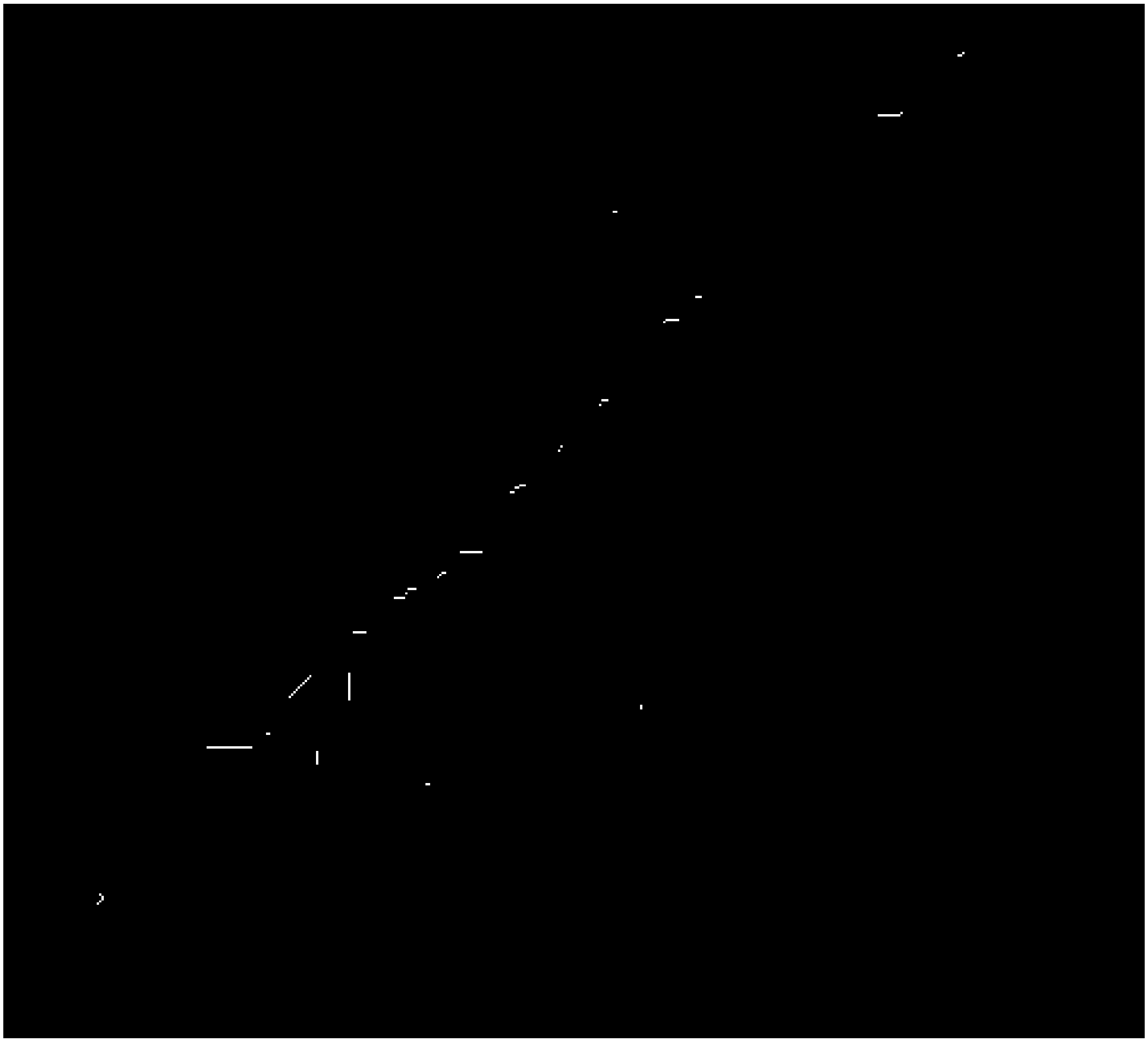

Figure 18: End-product of the filtering of the array shown in the previous Figure.