A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps

Martin A. Kraaijveld, Jianchang Mao, *Member, IEEE,* and Anil K. Jain, *Fellow, IEEE*

Abstract— **A nonlinear projection method is presented to visualize high-dimensional data as a two-dimensional image. The proposed method is based on the topology preserving mapping algorithm of Kohonen [13]–[16]. The topology preserving mapping algorithm is used to train a two-dimensional network structure. Then the interpoint distances in the feature space between the units in the network are graphically displayed to show the underlying structure of the data. Furthermore, we will present and discuss a new method to quantify how well a topology preserving mapping algorithm maps the high-dimensional input data onto the network structure. This will be used to compare our projection method with a well-known method of Sammon [28]. Experiments indicate that the performance of the Kohonen projection method is comparable or better than Sammon's method for the purpose of classifying clustered data. Another advantage of the method is that its time-complexity only depends on the resolution of the output image, and not on the size of the dataset. A disadvantage, however, is the large amount of CPU time required.**

I. Introduction

An important tool in exploratory data analysis is the projection of high-dimensional data onto a low-dimensional space to facilitate visual inspection of the data. This can provide better insight into the data, since clustering tendencies or a low intrinsic dimensionality in the data may become apparent from the projection. To preserve the inherent structure of the data as well as possible, the projection method should map the data faithfully onto the lower dimensional space. In general, this projection problem can be formulated as mapping a set of n vectors from an N-dimensional space onto an M-dimensional space, with $M < N$. Since the goal here is exploratory data analysis, we will be concerned with projections onto a two-dimensional plane ($M = 2$).

In this paper we will present a projection method that is based on the topology preserving mapping algorithm of Kohonen [13]–[16]. In the proposed method, which will be called the Kohonen projection method, the topology preserving mapping algorithm is used to project high-dimensional data onto a two-dimensional network structure. Then, with a new display technique, we will show how the inherent structure of the data can be visualized. Furthermore, a new method is presented to quantify how well a topology preserving mapping algorithm maps the data onto the network structure. This allows a quantitative evaluation of the quality of the mapping and thereby a comparison of topology preserving mapping algorithms with other projection methods. First, however, we will provide a short overview of some well-known projection methods.

In the literature on exploratory data analysis, several projection methods have been described. These projection methods try to preserve one of several criterion functions in the projection. Two important distinctions that can be made are whether the class labels of the data (if available) are used or not and whether the mapping is linear or nonlinear. This results in four possible types of projection algorithms which we will mention briefly here:

- *Unsupervised and Linear*: Among the linear projection methods for data without class labels, the eigenvector or Karhunen–Loeve projection [8] is probably the best known. Another powerful linear projection method is projection pursuit, developed by Friedman and Tukey [6].
- *Unsupervised and Nonlinear*: Sammon has presented a widely used algorithm in which the mean squared difference between the interpattern distances of points in the original space and in the projected space is minimized [28]. This generally results in a highly nonlinear mapping of the data. An approach that is somewhat related to Sammon's algorithm is multidimensional scaling [17] and [18]. Here a dataset often containing ordinal data is mapped onto a plane. A fundamentally different approach was presented by Wang *et al.* [31]. Their method projects the data onto the plane such that the minimum spanning tree of the data is preserved.
- *Supervised and Linear*: Discriminant analysis is a well-known procedure to project labeled data in a linear fashion [4]. In discriminant analysis, the ratio of the determinants of the between-class scatter matrix (S_B) and the within-class scatter matrix (S_w) is maximized. The solution is the space spanned by the eigenvectors corresponding to the largest eigenvalues of the matrix ($S_w^{-1} \cdot S_B$).
- *Supervised and Nonlinear*: An example of a nonlinear algorithm to project labelled data is presented in [7] and [8]. In this method, the coordinates of the points in the projected space are a function of the distance to

Manuscript received June 29, 1992; revised January 15, 1993, and accepted September 23, 1994. A short version of this paper has appeared in the *Proceedings of the 11th International Conference on Pattern Recognition,* This work was supported by the Dutch government as a part of the SPIN/FLAIR-DIAC project, by the Foundation of Computer Science in the Netherlands (SION), the Dutch Organization for Scientific Research (NWO), and by NSF Grants CDA 8806599 and IRI 8901513.

M. A. Kraaijveld is with the Pattern Recognition Group, Department of Applied Physics, Delft University of Technology, 2600 GA Delft, The Netherlands.

J. Mao and A. K. Jain are with the Department of Computer Science, Michigan State University, East Lansing, MI 48824 USA.

IEEE Log Number 9409160.

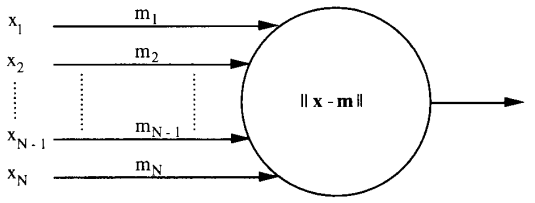


Fig. 1. The structure of a unit in a Kohonen network. Every unit computes the Euclidean distance between the N-dimensional input vector $\boldsymbol{x}$ and the weight vector $\boldsymbol{m}$.

the kth nearest neighbor of every point. In a number of applications it was shown that this projection preserves the underlying structure of the data. A second interesting method is described in [21]. Here the pairwise log-likelihood ratios of the points are used in the two-dimensional display. An unsupervised variant of this method was also presented.

The Kohonen projection method that is discussed in this paper falls in the category of nonlinear projection methods. Although the algorithm is basically unsupervised, many authors have demonstrated and used the topology preserving properties of the algorithm for problems in which the class labels are known, e.g., see [13]–[16] and the references therein. In this paper, however, we will assume that no information about the pattern class labels is available, and we will study how the Kohonen projection method compares to other unsupervised nonlinear projection methods. Category information of the data will only be used to evaluate the performance of the method.

The remainder of this paper is organized as follows. Sections II and III will present the Kohonen topology preserving mapping algorithm and Sammon's nonlinear projection algorithm and its variants. The Kohonen projection method is presented in Section IV together with the tools that are required for its evaluation. In Section V, a number of experiments will be presented which will be discussed in Section VI. Finally, the conclusions of this study are presented in Section VII.

II. The Kohonen Topology Preserving Mapping Algorithm

The topology preserving mapping algorithm of Kohonen is an iterative procedure for training a class of neural networks [13]–[16]. The learning procedure is unsupervised or self organizing and is used to train a network of units or neurons that are arranged in a low-dimensional structure (see Figs. 1–2). In this paper, a two-dimensional structure for the network is used, but in the literature the application of one and three-dimensional structures has frequently been described (e.g., see [15] and [26]).

The training of the network is initialized by assigning small random values to the weight vectors $\boldsymbol{m}$ of the units in the network. Each iteration in the learning process consists of three steps: the presentation of a randomly chosen input vector from the input space, the evaluation of the network, and an

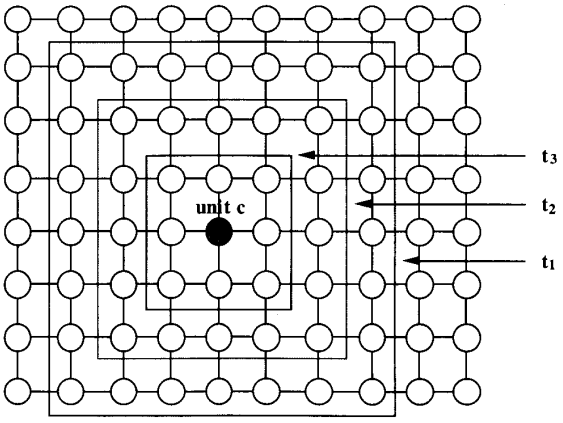


Fig. 2. A Kohonen network consisting of a two-dimensional array of units is shown. Every unit has the architecture as depicted in Fig. 1. On every step in the learning process, the unit c with the smallest Euclidean distance to the input-vector is determined. Then, all units within a certain neighborhood of unit c are updated according to the learning rule 2). The figure shows how the size of this neighborhood shrinks as a function of time. Early in the learning process, at t_1, a very large number of the units is updated on every step, whereas finally, at t_3, only a small fraction is updated.

update of the weight vectors. In the following, the iteration will be indexed by the time t. The weight vectors are updated according to the following procedure [13]–[16].

After the presentation of a pattern, the Euclidean distance between the input vector and the weight vector is computed for all units in the network. The unit with the smallest distance is marked as unit c

$$||\boldsymbol{x}(t) - \boldsymbol{m}_c(t)|| = \min_i (||\boldsymbol{x}(t) - \boldsymbol{m}_i(t)||). \tag{1}$$

In the following step, all units within a certain spatial neighborhood N_c around unit c are updated according to (see Fig. 2)

$$\boldsymbol{m}_i(t+1) = \begin{cases} \boldsymbol{m}_i(t) + \alpha(t)[\boldsymbol{x}(t) - \boldsymbol{m}_i(t)] & \text{if } i \in N_c(t) \\ \boldsymbol{m}_i(t) & \text{if } i \notin N_c(t). \end{cases} \tag{2}$$

The size of the neighborhood N_c is a function of time t and shrinks monotonically. The parameter $\alpha(t)$ is the step size of the adaptation of the weights and also shrinks monotonically with time. The update rule is closely related to the k-means clustering algorithm [20]. Like the k-means algorithm, it is the best matching unit (i.e., cluster center) which is moved a small step into the direction of the input vector. In the topology preserving mapping algorithm, however, a whole set of units are updated instead of a single unit. Since the units that are updated at every step are neighboring units in the network, there is a tendency that neighboring units in the network represent neighboring locations in the feature space. In other words, the topology of the data in the input space is preserved during the mapping. Clearly, when the intrinsic dimensionality of the data is higher than the dimensionality of the network,

the network will not be able to fully represent the structure of the data (see [15]). In that case, however, the network can be considered to be a low-dimensional representation of the data. It is this property of the algorithm that will be used in the Kohonen projection method described in this paper.

A slightly alternative formulation of the learning rule, which was used in our experiments, is the following [16]. Instead of updating all units in the neighborhood N_c identically, the update of a unit is weighed by a function of the distance to the best matching unit in the network. That is, when the coordinates of a unit in the network are given by $\boldsymbol{r}$ and the coordinates of the best matching unit by $\boldsymbol{r}_c$, a unit is updated according to

$$\boldsymbol{m}_i(t+1) = \boldsymbol{m}_i(t) + h_{ci}(t)[\boldsymbol{x}(t) - \boldsymbol{m}_i(t)] \tag{3}$$

where h_{ci} is a Gaussian weighting function

$$h_{ci}(t) = h_0(t) \exp\left(-\frac{||\boldsymbol{r}_i - \boldsymbol{r}_c||^2}{\sigma(t)^2}\right). \tag{4}$$

Here $h_0(t)$ and $\sigma(t)$ are chosen as suitably decreasing functions of time. In [16] it is discussed that the algorithm is relatively insensitive to the actual choice of these two parameters and the way in which they are decreased during the learning process. These findings are in accordance with our experiments, which are described in Section V.

Successful applications of this algorithm in speech recognition [14], robotics [26], AI [27], and many others are well known. For most of these applications, however, it is not always clear whether the algorithm offers any advantages over other competing methods. Rigorous theoretical analyses concerning various properties of this algorithm can be found in [24], [25], and [15]. These analyses study the convergence properties and the stability of the algorithm for some simple distributions of the data in the feature space. An important issue that is not addressed in these analyses, however, is the behavior of the algorithm when it is trained with a small amount of data. This is an important issue that is especially relevant in practical applications. A class of variants of the algorithm was presented as "learning vector quantization," e.g., see [16]. These are essentially modifications of the algorithm to use it for supervised learning problems.

III. Noniinear Projection with Sammon's Algorithm

Sammon's nonlinear projection algorithm [28] aims at minimizing an error measure that is a function of the differences of the interpoint distances in the original space and the interpoint distances in the projected space. Experimental results in [1] indicated that Sammon's algorithm has a performance that is superior over other algorithms. Therefore, we have chosen to compare the Kohonen projection method with Sammon's algorithm, rather than some other method. Moreover, two of the datasets that were used for the experiments in [1] are also used in the experiments that we describe in Section V, i.e., the IRIS data (dataset 5) and the 8OX data (dataset 4).

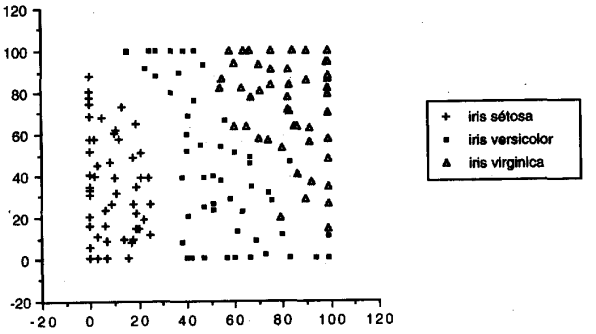


Fig. 3. A Kohonen network of 100 by 100 units was trained with the IRIS data (dataset 5 in Section V). When the class labels are assigned to the units after training, this projection clearly shows that the data is clustered and thereby demonstrates the topology preserving property of the algorithm.

When the distance between two patterns i and j is denoted d_{ij}^* in the original feature space, and d_{ij} in the projected space Sammon's algorithm minimizes the following measure of distortion of the projection

$$E = \frac{1}{\sum\sum_{i<j} d_{ij}^*} \sum\sum_{i<j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}. \tag{5}$$

This is an optimization problem that can be solved with a suitable optimization technique, as the gradient descent procedure proposed by Sammon [28]. Since for every step in an iteration of Sammon's algorithm $n(n-1)/2$ distances have to be computed, the algorithm quickly becomes impractical for large amounts of data. Therefore, numerous authors have proposed methods to lower the time complexity of the algorithm, e.g., see [3], [23], [19], and [1]. Furthermore, a number of variants of the algorithm have been published. Among these are the use of different metrics [32], [12], different optimization criteria [29] or different optimization methods [12].

IV. The Kohonen Projection Method

As discussed in Section II, the topology preserving mapping algorithm can be used to project data onto the low-dimensional network structure. An example is presented in Fig. 3. In this figure the well-known IRIS data (dataset 5 in Section V) is used to train a network. The figure shows the labelling of the units in a large Kohonen network (100 × 100) after the learning process. From the fact that the three classes are well separated in the network plane, it can be decided that the classes are clustered. It is important to note, however, that the structure of the data can only be perceived through this labeling of the units. Therefore, for problems for which no class labels are available, this procedure will not work.

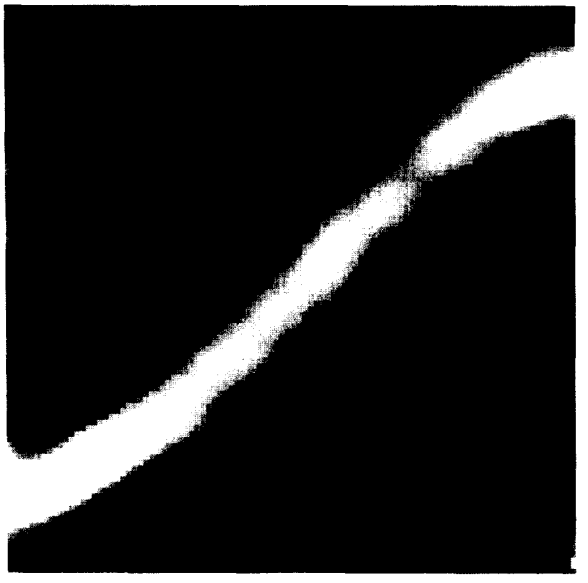


Fig. 4. Projection image of dataset 1—10 dimensional separated normal clusters.

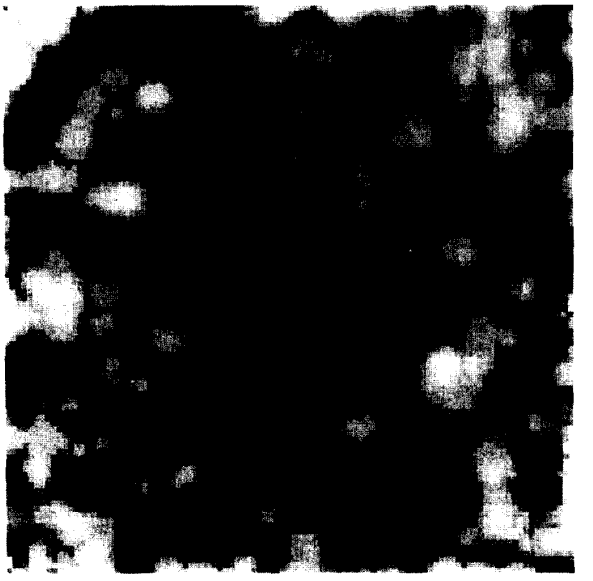


Fig. 5. Projection image of dataset 9—Uniform distributed 10 dim. data.

The solution to the labeling problem that is presented in this paper has two components. In the first place, a rather large network is used. For the experiments reported in this paper, a two-dimensional network of 100 by 100 units was used. The second step of our solution is to display the network as an image, whereby every unit corresponds to a pixel. The gray value of each pixel is determined by the maximum distance in the feature space of the corresponding unit to its four neighbors (East, West, North, and South) in the network. The larger the distance, the lighter the gray value is.

An example of this method is presented in Fig. 4. In this case, two 10-dimensional Gaussian distributed clusters (dataset 1) were used to train the network. It is apparent from the projection image that there are two dark regions, corresponding to regions where the units are very close in the feature space, and one bright line, which corresponds to the empty region between the two clusters. In each dark region the units are relatively close, so the distance in the feature space of a unit to its four neighbors in the network is small. For all units in the bright region, however, there is at least one neighboring unit that is far(ther) away, so the corresponding gray value is higher. Note that the network has only a two-dimensional topology and is therefore not capable of fully capturing the 10-dimensional nature of the individual clusters. The image clearly shows, however, that the dataset consists of two well-separated clusters. It is illustrative to compare the result of Fig. 4 with that of Fig. 5. In the latter case the dataset consists of uniformly distributed 10-dimensional data (dataset 9). Since there are hardly any clustering tendencies in this dataset, it is interesting to notice that there is no apparent structure in the corresponding projection image. From these results it can be concluded that the proposed projection method works in principle. The questions that now remain to be answered are how can the quality of the mapping be quantified and how to relate its performance to the performance of other projection methods. Therefore, the second contribution of this paper consists of such a quantification method.

One of the problems that arises in the evaluation of the Kohonen projection method is that there is no direct notion of interpoint distances in the projection. This is different from all other mapping methods where the data are projected directly onto a lower dimensional space. In this projected space, the distances are easily computed, which facilitates the direct usage of an error measure like Sammon's distortion measure in (5). In our approach, distances are displayed indirectly by the gray value, and the only distances that are displayed are the distances between the four immediate neighbors. To be able to evaluate the new projection method, it is necessary to define a distance measure in the network plane. Therefore, we will define a metric that is essentially based on a graph searching technique; see Fig. 6. Its functionality and implementation closely resembles that of the gray value weighted distance transform as described by Verbeek and Verwer [30]. First, however, we need some definitions.

Definition 1: The distance between two units (see Fig. 6);

- The distance d_{ij} between two eight-connected neighboring units i and j in the network plane is defined as the Euclidean distance d^*_{ij} of the units in the feature space.
- The distance d_{ij} between two nonneighboring units i and j in the network plane is defined as the minimum of the summed distances between neighboring units over all possible eight-connected paths in the network plane from unit i to unit j.

An informal interpretation of these definitions is that the distance between two points in the image is determined by

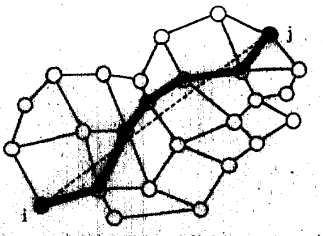


Fig. 6. The computation of the distance between two units i and j. As in Fig. 2, the circles represent the units of the network; the black lines show the neighboring relations in the network plane. An example is given of a part of a network in a two-dimensional feature space; the location of the units corresponds to the location of the weight vectors in the feature space. The dashed line between i and j denotes the true distance $d^*_{i,j}$ between the units, whereas the gray line represents the approximation $d_{i,j}$ of the true distance by taking discrete steps between neighboring units in the network plane. The gray line is found by searching for the shortest path among all possible paths between units i and j. Note that the gray line consists of steps between four connected neighbors, as well as steps between diagonal neighbors.

integrating the gray values over all possible paths between the two points and then selecting the path with the lowest sum. It is important to note that this informal interpretation is not exactly equal to the formal definition. This is because the definitions above are explicitly defined in the network plane and not in the image. The differences are based on two related facts. In the first place, the projection image only shows information about the distance between a unit and its four neighbors, whereas the estimate of the distance makes use of the distances to eight neighbors. Clearly, this improves the estimates of the path lengths. Second, the projection image effectively shows the distance to only one neighbor for every unit (i.e., the farthest), whereas our distance estimate makes use of the distance to eight neighbors. For the projection image this is advantageous since this increases the contrast in the projection image. The estimation of the distances, however, clearly benefits from taking more neighbors into account.

Now that we are able to compute the distance between two units, we can define the distance between two patterns in the network plane.

Definition 2: The distance between two patterns in the network plane is defined as the distance between the two corresponding closest units.

Now all the necessary tools are available to compare the projection methods. Since for almost all datasets that were used in the experiments the class labels were available, we have chosen to use the following evaluation criteria (see also [1]):

- The Sammon error measure; see (5). This indicates how well the interpattern distances have been preserved in the projection.
- The difference of the performance of the nearest neighbor classifier in the original and the projected space. This measures how well "local" information has been preserved in the projection.
- The difference of the performance of the nearest mean classifier (also known as the minimum distance classifier) in the original and the projected space. This indicates how well "global" information has been preserved in the projection.

One might wonder whether it is computationally feasible to compute distances over all possible paths in the network plane and to repeat this for all $O(n^2)$ interpoint distances. Clearly, a brute force solution would require a prohibitive amount of CPU time. Our implementation is based on the following approach. A point i from the dataset is projected on the network; i.e., the closest unit is determined. Then, as in the implementation of the gray value weighted distance transform [30], all other units in the network are labeled with the shortest distance to that unit. The time complexity of this process is roughly in the order of the number of units in the network. The distance to a point j from the dataset is now simply found by projecting the point onto the network and to read the distance from the label in the corresponding closest unit. The latter operation takes an amount of time that is negligible compared to the process of labelling the units with the distance. The time complexity of the complete procedure is therefore roughly in the order of the number of units in the network times the number of points in the dataset. Although this still results in considerable requirements with respect to the CPU time, the experiments indicated that the complete evaluation procedure only took 10% of the time that was spent in training the network. From that point of view, the proposed evaluation methods are indeed computationally feasible.

A final remark is that an alternative use of these evaluation methods is to quantify how well a Kohonen network has been able to map the data onto the network structure. For example, variants of the algorithm can quantitatively be compared in this way.

V. Experiments

To test the projection method with the criteria mentioned above, a number of experiments were conducted. In this section the datasets, the experimental procedures and the results will be discussed.

A. Datasets

To test the performance of the projection method, a large variety of datasets was used. Among these are four artificial datasets and five datasets consisting of real data.

- *Dataset 1*: Artificial dataset consisting of two standard normally distributed clusters of 500 patterns each, in a 10-dimensional space. The means of the clusters are (-1, -1, -1, $\cdots$, -1) and (+1, +1, +1, $\cdots$, +1) and the covariance matrix of both clusters is equal to the identity matrix. The Bayes error for the two distributions is only 0.078%, so the two clusters are very well separated in the feature space.
- *Dataset 2*: Artificial dataset consisting of two elongated clusters of 500 patterns each, in a nonlinear two-dimensional subspace of the three-dimensional feature space; see Fig. 7. This dataset was generated with the

following pseudo code:

```
Class A :

    theta = Pi * ( - 0.5 + random_unif( ));

    x = 0.5 * cos(theta) + 0.025
        * random_gauss( );
    y = 0.5 * sin(theta) + 0.025
        * random_gauss( );
    z = sin(2 * x) * cos(2 * y) + 0.025
        * random_gauss( );

Class B :

    theta = Pi * (0.5 + random_unif( ));

    x = 0.25 + 0.5 * cos(theta) + 0.025
        * random_gauss ( );
    y = 0.5 + 0.5 * sin(theta) + 0.025
        * random_gauss( );
    z = sin(2 * x) * cos(2 * y) + 0.025
        * random_gauss( );
```

This dataset was used because it is (almost) intrinsically two dimensional. It is, therefore, to be expected that it perfectly maps to the two-dimensional structure of the network.

- *Dataset 3*: Artificial dataset consisting of uniformly distributed data on the surfaces of two three-dimensional spheres: a large sphere at (0, 0, 0) with radius one, and a small sphere within the large sphere at (0, 0, 0.2) with radius 0.1. This dataset was chosen because it is particularly difficult for most clustering algorithms [9].
- *Dataset 4*: Real dataset consisting of the well-known 8OX hand printed character data. It consists of 45 patterns in an eight-dimensional feature space. The data consists of three classes (the characters "8," "O," and "X") and is very sparsely distributed in the feature space [9].
- *Dataset 5*: Real dataset consisting of the well-known IRIS dataset [5]. It consists of 150 patterns in three classes in a four-dimensional feature space.
- *Dataset 6*: Real dataset extracted from the range image of a polyhedral object; see Fig. 8. Of all the 13 633 pixels in the range image, the z coordinate and the (three component) surface normal vector was computed. In [9] it was shown how range data can be segmented with the help of a clustering algorithm in this feature space. Here we use the Kohonen projection method to visualize the clustering tendencies of the dataset.
- *Dataset 7*: Real dataset extracted from a 256 $\times$ 256 image with four textures synthesized by four different Gaussian Markov random fields; see Fig. 9. The dataset contains 15 multi-resolution SAR (i.e., simultaneous autoregressive) model features for every pixel [22]. The x and the y coordinates of every pixel were included as two additional features. The total number of patterns in the dataset was 4000.

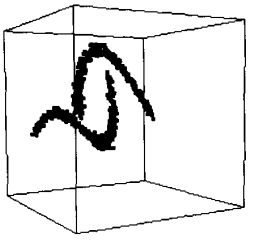


Fig. 7. Dataset 2: two elongated clusters in a nonlinear two-dimensional subspace in the three-dimensional feature space.

- *Dataset 8*: Real dataset extracted from a composite 512 $\times$ 512 image with 16 textures from the Brodatz book [2]; see Fig. 10. The image was filtered with 20 Gabor filters, giving 20 features for every pixel [10]. The x and the y coordinate of every pixel were included as two additional features. The total number of patterns in the dataset was 16 000.
- Dataset 9: Artificial dataset consisting of 1000 uniformly distributed patterns in a 10-dimensional cube. This dataset exhibits almost no clustering tendency and is, therefore, expected to result in a projection image with hardly any structure.

B. Experimental Procedures

The starting point for all the experiments was a Kohonen network consisting of 100 by 100 units. The choice for this size of the network was based on the available amount of memory in the computers available to us. The network should preferably be as large as possible, however, to provide the user with the largest possible resolution in the projection image. Every dataset was used to train 10 networks with the same architecture but with different initial weights, so that statistics about the performance of the network could be collected. The simulations were based on a custom made program in C and were performed on SUN Sparc II workstations.

The parameters of the Kohonen learning algorithm were based on a few initial experiments with some of the datasets. After the selection of the parameter values, the same values were used for all the datasets and for all the experiments. They were set to the following values. The initial value of the parameter controlling the step size of the updates h_0 (0) was 0.05, see (3). After every update of the weights (i.e., after the presentation of a pattern to the network), $h_0(t)$ was decreased with a factor 0.9999, with a minimum value of 0.0001. The width of the kernel weighing the update of the units, σ (0), was initially set to 66.666 and $\sigma(t)$ was also decreased by a factor 0.9999, with a minimum of 1.0. In advance of the training procedure, the order of the patterns in the datasets was randomized, and then all the patterns were cyclically presented to the network. The training of the network was terminated after 100 000 weight vector updates. After the training phase of the network, the Sammon distortion and the error rates of the nearest neighbor classifier and the nearest mean classifier were computed with the leave-one-out method. The classifiers were implemented by projecting the dataset or the class means onto

the network structure. Special care was taken to deal with the problem of multiple data points mapping onto the same unit. In that case the unit was labeled with the majority class label of the data points that were projected onto the unit. As the distance measure allows the computation of the distances and thereby also of neighboring relations, a new sample could then easily be labelled by searching for the nearest unit with a class label. To limit the amount of CPU time required for datasets larger than 1000 patterns, the estimates of the performance of both classifiers were based on a randomly selected subset of 1000 patterns.

To compare the Kohonen projection method with an alternative method, all experiments were repeated with Sammon's algorithm. Every dataset was also projected 10 times with Sammon's algorithm and statistics about the performance were collected. Since the CPU and memory requirements of Sammon's algorithm become prohibitive for large amounts of data, the datasets that were larger than 1000 patterns were replaced by a subset consisting of 1000 randomly selected patterns. The step size for the gradient descent procedure in Sammon's algorithm (i.e., the Magic Factor, see [28]) was chosen as 0.3. After the projection, the performance of the nearest neighbor classifier and the nearest mean classifier were computed with the leave-one-out method.

C. Results

The results of the experiments are summarized in Tables I–III. Since the data in dataset 9 (the uniformly distributed noise in a cube) was not labeled, the estimates of the nearest neighbor and nearest mean performance are omitted for this dataset.

In Figs. 4, 5, and 11–17, the projection images of the Kohonen projected data are shown. When the class labels of the data are available, a particularly good display technique can be derived by showing the labels in a color overlay on the projection image. For comparison, the Sammon's projection of the IRIS data is shown in Fig. 18.

VI. Discussion

From the projection images shown in Figs. 4, 5, and 11–17, it can be seen that they indeed visualize the true structure of the data. The best examples of this are found in Fig. 4 (dataset 1) and Fig. 5 (dataset 9), respectively, corresponding to a well-clustered dataset and a dataset without any clustering tendency. Fig. 4 is indeed very structured, whereas Fig. 5 shows very little or no structure. Moreover, for datasets 2, 3, 5, 6, and 7 it is clear that there are indeed clustering tendencies in the data. Some limitations of the method can be found in Figs. 13 and 17. In Fig. 13 the problem is caused by the very sparse nature of the dataset. The image contains roughly as many dark regions as there are patterns in the dataset (i.e., 45). This indicates that every pattern is considered to represent a cluster by itself or, in other words, that there is no clustering tendency detected in the data. This is in accordance, however, with the results of other projection algorithms on this dataset (e.g., see [9]). A second potential problem is found in Fig. 17. Here, a large number of clusters "struggle" for the limited available space in the image. Probably, a better result could be obtained by using a larger network.

From the quantitative results reported in Section V, it is apparent from Table I that Sammon's algorithm performs significantly better in preserving the interpoint distances than the Kohonen algorithm. This is not surprising, since the Kohonen algorithm does not aim at minimizing Sammon's error measure. The exception is dataset 2, which represents the best possible case for the Kohonen algorithm: the data is clustered and has an intrinsic dimensionality that is equal to the dimensionality of the network structure. Another interesting result is found for dataset 9, which corresponds to the worst possible case for any projection algorithm. Here, the data has no clustering tendency at all and has an intrinsic dimensionality that is higher than the dimensionality of the network structure. For the Kohonen projection method, this indeed results in an extremely high distortion. For Sammon's algorithm, however, the structure of this data caused the algorithm not to converge.

The results presented in Tables II and III show that the Kohonen projection method varies between slightly better to significantly better than Sammon's algorithm in preserving the performance of the nearest neighbor classifier and the nearest mean classifier. This implies that for applications in which the projected data have to be classified afterwards, the Kohonen algorithm is to be preferred over Sammon's algorithm. An example of such an application is the speech recognition system as described by Kohonen [14]. Also, it is apparent that the performance of the nearest neighbor classifier is slightly better preserved than the performance of the nearest mean classifier. This can be explained by the fact that the topology preserving mapping algorithm is still based on localized updates of the units.

A subtle difference between the two projection algorithms is when they are used for the projection of points that are not part of the original dataset. In the Kohonen projection method, an unforeseen point is projected by searching for the closest unit in the network. Then, its projection can be visualized by highlighting the corresponding pixel in the projection image. When the number of units in the network is larger than the size of the dataset, the Kohonen algorithm has the advantage that the network interpolates between the points of the dataset. This provides a good estimate of the projection of a new point, where accuracy can be controlled by the size of the network. For Sammon's algorithm, the projection of an unforeseen point can analogously be accomplished by searching for the nearest neighbor of the new point in the dataset. Then, as an estimate of the location of the new point in the projected space, the projection of its nearest neighbor can be used. When the size of the original dataset is small, the accuracy of this procedure can potentially be increased by averaging over some of the projected nearest neighbors. This involves an additional procedure, however, that is not implicit in the projection method.

Another issue that needs discussion is the speed of the Kohonen projection method, since it might prevent its practical

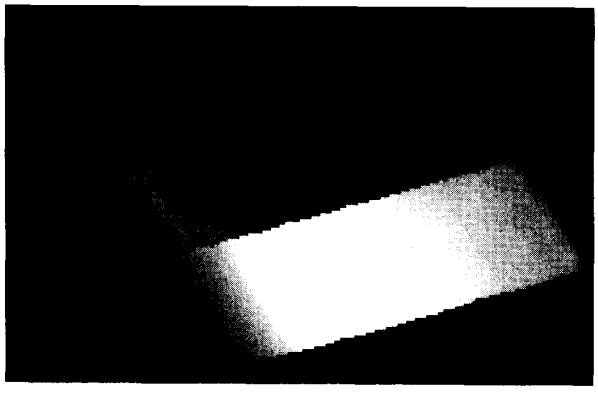


Fig. 8. Dataset 6: The range image of the polyhedral object. For all 13 633 pixels in this image, the z-coordinate and the three component surface normal vector were computed. These were used as four features for training a Kohonen network.

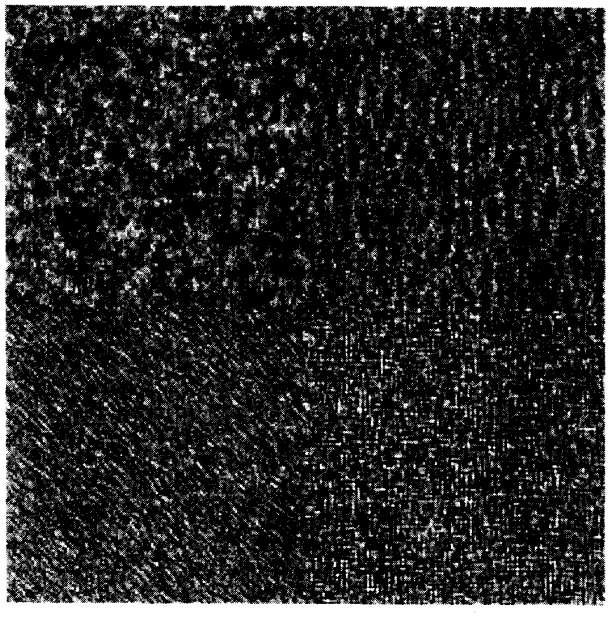


Fig. 9. Dataset 7: An image containing four synthesized textures. The textures were generated by four different Gaussian–Markov random fields.

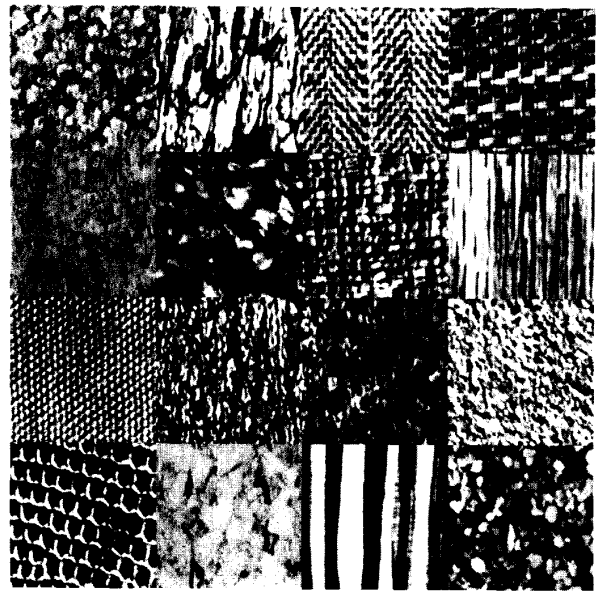


Fig. 10. Dataset 8: A 512 × 512 image containing 16 textures from the Brodatz book [2].

Fig. 11. Projection image of dataset 2—two elongated clusters in 3D space.

application. As Sammon's algorithm was in 1969, the Kohonen projection method is, in its current form, not very practical. In the implementation that was used for the experiments, every projection in Section V took up to tens of hours of CPU time. Clearly, this is not fast enough for interactive use. Since 90% of the CPU time was spent in training the network with the Kohonen topology preserving mapping algorithm, the projection method can be speeded up by using faster variants of the Kohonen algorithm. The issue of investigating faster variants of the Kohonen algorithm, however, was considered not to be within the scope of this paper. With the regular Kohonen algorithm in our implementation, we estimate that the speed can possibly be improved by a factor 10 by using other parameters for the Kohonen algorithm (e.g., lowering the number of iterations to 25 000 or 50 000), and by optimizing the simulator. Then, by using a computer that is 10 times faster, the CPU-time could be brought back to tens of minutes instead of tens of hours. Due to the parallel nature of the Kohonen algorithm, another promising way to speed up the projection is by using parallel computers or special purpose hardware. This may bring the projection time back from minutes to

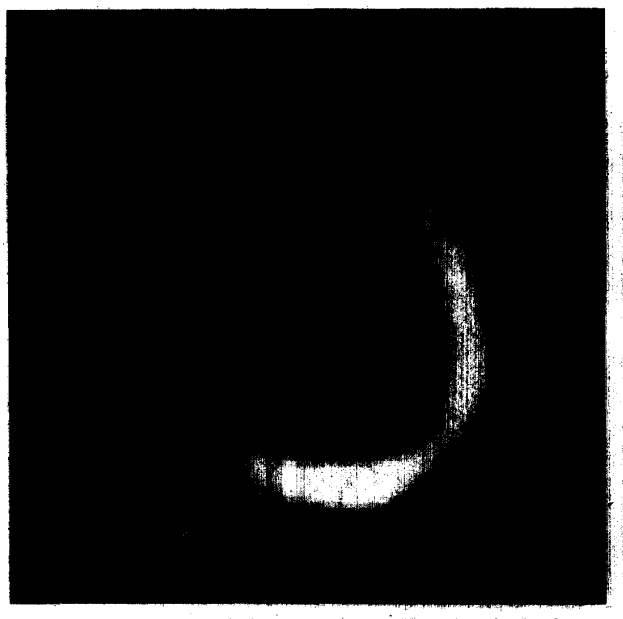


Fig. 12. Projection image of dataset 3—small sphere within large sphere.

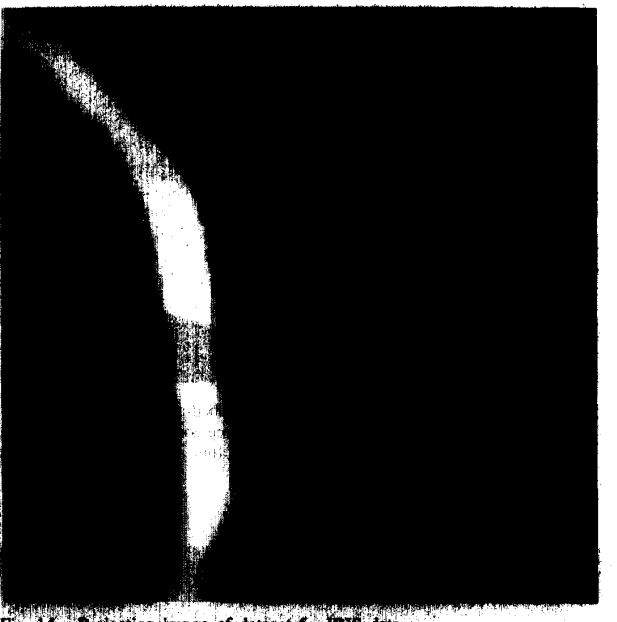


Fig. 14. Projection image of dataset 5—IRIS data.

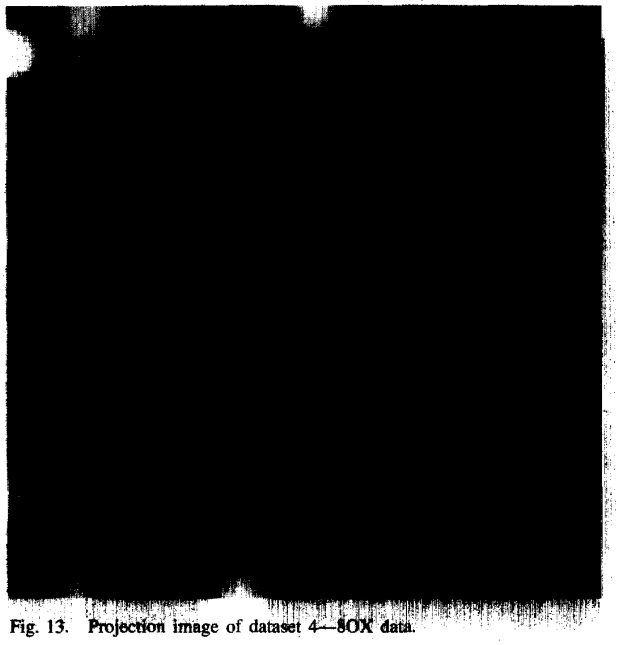


Fig. 13. Projection image of dataset 4—BOX data.

Fig. 15. Projection image of dataset 6—range image data.

seconds. Another interesting difference between the algorithms is that the CPU time for Sammon's algorithm is proportional to the square of the number of samples in the dataset, whereas the Kohonen projection method is linear in the number of units in the network. Therefore, by choosing the resolution of the projection image one can directly influence the required amount of CPU time.

A final remark is that, as can be seen in Fig. 6, the approximation of distances by taking discrete steps in the network plane clearly results in an overestimate of the distance. This suggests that this discretization effect could be corrected by multiplying the estimated distance with a correction factor slightly smaller than one. Although it is doubtful that a universal constant exists which is optimal for all network sizes and all probability distributions, some theoretical work could be done on determining the value of this constant for certain probability distributions and network sizes. Also, empirical research might indicate that a constant

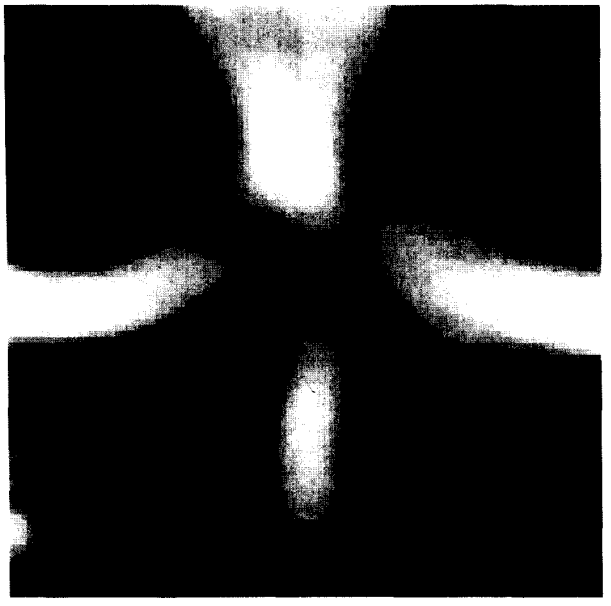


Fig. 16. Projection image of dataset 7—4 textures data.

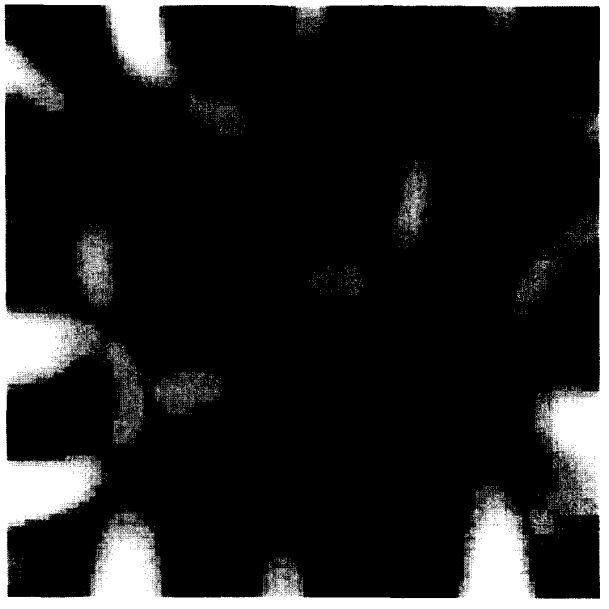


Fig. 17. Projection image of dataset 8—16 textures data.

exists which improves the results in a number of realistic applications.

VII. Conclusions

The nonlinear projection method that is presented in this paper is based on three ideas, of which two ideas are contributions of this paper. In the first place, the well-known Kohonen topology preserving mapping algorithm is used to project high-dimensional data onto a two-dimensional network structure. Secondly, the structure of the data is visualized by mapping the network onto a two-dimensional image. In this image, the gray value of every pixel (i.e., unit) is proportional to the distance to the farthest neighbor in the network plane. Finally, a technique based on the gray value weighted distance transform [30] facilitates the definition of a metric in the network plane and thereby enables a quantitative evaluation of the algorithm. The experimental results indicate that the Kohonen projection method has a performance that is comparable or better than Sammon's algorithm for the purpose of classifying clustered data. For the purpose of preservation of the interpoint distances, however, Sammon's algorithm performs better. Although the current implementation is very slow, the algorithm can be speeded up significantly by mapping the algorithm onto a parallel computer. Furthermore, the time complexity of the proposed algorithm depends on the resolution of the projection image, and not on the number of samples in the dataset. A final remark is that the use of the metric in the network plane facilitates a quantitative evaluation of various topology preserving mapping algorithms.

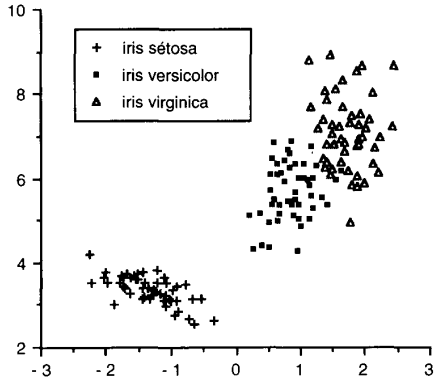


Fig. 18. The projection of the IRIS data (dataset 5) with Sammon's algorithm.

TABLE I
The Average Sammon Distortion (in %) and Its Standard Deviation (in Brackets)

Set	Description	Sammon distortion	
		Kohonen projection	Sammon's projection
1	10 dim. separated normal clusters	**34.6** (1.2)	**6.0** (0.1)
2	2 elongated clusters in 3D space	**0.51** (0.0)	**2.7** (1.2)
3	small sphere within large sphere	**29.9** (1.2)	**7.5** (0.7)
4	8OX data	**49.3** (2.2)	**5.3** (0.4)
5	IRIS data	**3.4** (0.4)	**0.6** (0.1)
6	range image data	**6.9** (0.8)	**2.1** (1.0)
7	4 textures data	**33.4** (6.0)	**5.3** (0.1)
8	16 textures data	**42.0** (6.9)	**5.7** (0.1)
9	Uniform distributed 10 dim. data	**95.4** (2.6)	did not converge

Acknowledgment

A large part of the software that was used for the experiments was developed in close cooperation with W. F. Schmidt of the Pattern Recognition Group, Department of Applied

TABLE II
THE AVERAGE ERROR OF THE NEAREST NEIGHBOR CLASSIFIER (IN %), ESTIMATED WITH THE LEAVE-ONE-OUT METHOD AND ITS STANDARD DEVIATION (IN BRACKETS)

Set	Description	Classification Error		
		Using input feature space	Kohonen projection	Sammon's projection
1	10 dim. separated normal clusters	0.2	0.3 (0.1)	0.3 (0.1)
2	2 elongated clusters in 3D space	0.0	0.0 (0.0)	0.3 (0.4)
3	small sphere within large sphere	0.0	0.0 (0.0)	0.3 (0.2)
4	8QX data	4.4	4.9 (2.2)	11.8 (4.0)
5	IRIS data	4.0	4.1 (0.7)	5.5 (1.8)
6	range image data	4.1	7.3 (1.2)	7.3 (0.6)
7	4 textures data	1.8	2.8 (0.4)	5.3 (0.2)
8	16 textures data	3.5	4.6 (0.9)	27.0 (2.5)

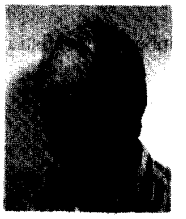
TABLE III
THE AVERAGE ERROR OF THE NEAREST MEAN CLASSIFIER (IN %), ESTIMATED WITH THE LEAVE-ONE-OUT METHOD AND ITS STANDARD DEVIATION (IN BRACKETS)

Set	Description	Classification Error		
		Using input feature space	Kohonen projection	Sammon's projection
1	10 dim. separated normal clusters	0.1	3.1 (3.5)	0.1 (0.1)
2	2 elongated clusters in 3D space	13.3	13.5 (0.1)	13.4 (1.2)
3	small sphere within large sphere	39.0	47.4 (4.3)	30.6 (1.3)
4	8QX data	4.4	12.9 (2.2)	16.0 (5.4)
5	IRIS data	8.0	6.6 (1.8)	7.7 (1.3)
6	range image data	15.2	15.8 (1.7)	21.2 (3.6)
7	4 textures data	1.9	3.2 (0.8)	4.5 (0.2)
8	16 textures data	6.5	8.8 (1.4)	28.4 (0.9)

Physics, Delft University of Technology. Dr. R. P. W. Duin and Dr. A. M. Vossepoel, also of the Pattern Recognition Group, Department of Applied Physics, Delft University of Technology, are gratefully acknowledged for some interesting discussions on Sammon's algorithm and gray value weighted distance transforms.

REFERENCES

[1] G. Biswas, A. K. Jain, and R. C. Dubes, "Evaluation of projection algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 6, pp. 701–708, Nov. 1981.
[2] P. Brodatz, *Textures—A Photographic Album for Artists and Designers*. New York: Dover, 1966.
[3] C. L. Chang and R. C. T. Lee, "A heuristic relaxation method for nonlinear mapping in cluster analysis," *IEEE Trans. Syst. Man Cybern.*, vol. 3, pp. 197–200, Mar. 1973.
[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[5] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, pp. 178–188, 1936.
[6] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. 23, pp. 881–890, Sept. 1974.
[7] K. Fukunaga and J. M. Mantock, "A nonparametric two-dimensional display," in *Proc. 1980 Int. Conf. Syst., Man Cybern.*, Cambridge, MA, Oct. 8–10, 1980.
[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
[9] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[10] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, pp. 1167–1186, 1991.
[11] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, N.Y. 220, pp. 671–680, 1983.
[12] R. W. Klein and R. C. Dubes, "Experiments in projection and clustering by simulated annealing," *Pattern Recognition*, vol. 22, no. 2, pp. 213–220, 1989.
[13] T. Kohonen, "Clustering, taxonomy, and topological maps of patterns," in *Proc. Sixth Int. Conf. Pattern Recognition*, Munich, Germany, pp. 114–128, 1982.
[14] ——, "The 'neural' phonetic typewriter," *Comput.*, vol. 21, pp. 11–22, Mar. 1988.
[15] ——, *Self Organization and Associative Memory*, 3rd ed. Heidelberg, Germany: Springer-Verlag, 1989,
[16] ——, "The self organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sept. 1990.
[17] J. B. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, pp. 115–129, 1964.
[18] J. B. Kruskal, "Multidimensional scaling and other methods for discovering structure," in *Statistical Methods for Digital Computers*, K. Enslein, A. Ralston, and H. S. Wilf, Eds. New York: Wiley, 1977, pp. 296–339.
[19] R. C. T. Lee, J. R. Slagle, and H. Blum, "A triangulation method for the sequential mapping of points from N-space to two-space," *IEEE Trans. Comput.*, vol. 27, pp. 288–292, Mar. 1977.
[20] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probability*, 1967, pp. 281–297.
[21] J. M. Mantock and K. Fukunaga, "A two-dimensional display for multiclass multivariate data," *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, Eds. Amsterdam: North-Holland, 1980, pp. 361–368.
[22] J. C. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.
[23] C. B. Pykett, "Improving the efficiency of Sammon's nonlinear mapping by using clustering archetypes," *Electron. Lett.*, vol. 14, pp. 799–800, 1978.
[24] H. Ritter and K. Schulten, "On the stationary state of Kohonen's self-organizing sensory mapping," *Biological Cybern.*, vol. 54, pp. 99–106, 1986.
[25] ——, "Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection," *Biological Cybern.*, vol. 69, pp. 59–71, 1989.
[26] H. J. Ritter, T. M. Martinetz, and K. J. Schulten, "Topology conserving maps for learning visuo-motor-coordination," *Neural Networks*, vol. 2, pp. 159–168, 1989.
[27] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biological Cybern.*, vol. 61, pp. 241–254, 1989.
[28] J. W. Sammon Jr., "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. 18, pp. 491–409, May 1969.
[29] B. Schachter, "A nonlinear mapping algorithm for large data sets," *Comput. Graphics Image Process.*, vol. 7, pp. 271–278, 1978.
[30] P. W. Verbeek and B. J. H. Verwer, "Shading from shape, the Eikonal equation solved by gray-weighted distance transform," *Pattern Recognition Lett.*, vol. 11, pp. 681–690, Oct. 1991.
[31] D. K. Wang, R. B. Urquhart, and J. E. S. MacLeod, "The equal-angle spanning tree mapping: A sequential method for projecting from h-space to 2-space," *Pattern Recognition Lett.*, vol. 2, pp. 69–73, 1983.
[32] I. White, "Comment on 'A nonlinear mapping for data structure analysis'," *IEEE Trans. Comput.*, Feb. 1992, pp. 220–221.



Martin A. Kraaijveld received the M.Sc. degree in electrical engineering in 1986, from Delft University of Technology in Delft, The Netherlands, and the Ph.D. degree in applied physics from Delft University in 1993.

Since May 1993 Dr. Kraaijveld has been working on neural networks at the General Research Department of the Exploration and Production Laboratory of Shell Research in the Netherlands. His research interests include theoretical and practical aspects of statistical pattern recognition and neural networks, image processing, computer vision, computer graphics and special purpose computer architectures.

Jianchang Mao (S'90–M'94) received the B.S. degree in physics in 1983 and the M.S. degree in electrical engineering in 1986, from East China Normal University, Shanghai, P.R. China. He received the Ph.D. degree in computer science from Michigan State University, East Lansing, MI, in 1994.

Dr. Mao was a graduate research assistant in the Computer Science Department during the period of four years at Michigan State University. During the summer of 1993, he worked at the Xerox Palo Alto Research Center on document image processing. Since January 1994, he has worked with the IBM Almaden Research Center where he spent two months in 1993 as a student co-op. His research interests include pattern recognition, neural networks, OCR, document image processing, computer vision, and parallel computing.

Anil K. Jain (M'72–SM'86–F'91) received a B.Tech. degree in 1969 from the Indian Institute of Technology, Kanpur, India, and the M.S. and Ph.D. degrees in electrical engineering from Ohio State University, in 1970 and 1973, respectively.

He joined the faculty of Department of Computer Science at Michigan State University in 1974 and is a University Distinguished Professor at Michigan State University. He served as Program Director of the Intelligent Systems Program at the National Science Foundation (1980–1981) and has held visiting appointments at Delft University of Technology, The Netherlands, Norwegian Computing Center, Oslo, and Tata Research Development and Design Center, Pune, India. He has also been a consultant to several industrial, government and international organizations. His current research interests are computer vision, image processing, artificial neural networks, and pattern recognition.

He has published papers on the following topics: statistical pattern recognition, artificial neural networks, exploratory pattern analysis, remote sensing, Markov random fields, texture analysis, interpretation of range images, and 3-D object recognition. Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He received the best paper awards in 1987 and 1992 and certificates for outstanding contributions in 1976, 1979, and 1993 from the Pattern Recognition Society.

Dr. Jain is the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and serves on the editorial boards of the *Pattern Recognition Journal, Pattern Recognition Letters, Journal of Intelligent Systems,* and *Journal of Mathematical Imaging and Vision*. He is the co-author of *Algorithms for Clustering Data*, (Prentice-Hall, 1988), has edited the book *Real-Time Object Measurement and Classificaton*, (Springer-Verlag, 1988), and co-edited the books, *Analysis and Interpretation of Range Images*, (Springer-Verlag, 1990), *Markov Random Fields: Theory and Applications*, (Academic Press, 1993), *Statistical Pattern Recognition and Artificial Neural Networks: Old and New Connections*, (North-Holland, 1991), and *Three-Dimensional Object Recognition Systems*, (Elsevier, 1993).

Dr. Jain was the General Chairman of the IEEE Computer Society Workshop on Interpretation of 3-D Scenes, Austin (1989), Co-Chairman of the Eleventh International Conference on Pattern Recognition, The Hague (1992), Program Director of the NATO Advanced Research Workshop on Real-Time Object Measurement and Classification, Maratea (1987), and co-directed NSF supported Workshops on Challenges of Computer Vision: Future Research Directions, Maui (1991), Theory and Applications of Markov Random Fields, San Diego (1989) and Range Image Understanding, East Lansing (1988). Dr. Jain received the Distinguished Faculty Award from Michigan State University in 1989 and served as the Distinguished Visitor of the IEEE Computer Society during 1988–1990.