

# Nonlinear Embedded Map Projection for Dimensionality Reduction

Simone Marinai, Emanuele Marino, and Giovanni Soda

Dipartimento di Sistemi e Informatica - Università di Firenze  
Via S.Marta, 3 - 50139 Firenze - Italy

**Abstract.** We describe a dimensionality reduction method used to perform similarity search that is tested on document image retrieval applications. The approach is based on data point projection into a low dimensional space obtained by merging together the layers of a Growing Hierarchical Self Organizing Map (GHSOM) trained to model the distribution of objects to be indexed. The low dimensional space is defined by embedding the GHSOM sub-maps in the space defined by a non-linear mapping of neurons belonging to the first level map. The latter mapping is computed with the Sammon projection algorithm.

The dimensionality reduction is used in a similarity search framework whose aim is to efficiently retrieve similar objects on the basis of the distance among projected points corresponding to high dimensional feature vectors describing the indexed objects.

We compare the proposed method with other dimensionality reduction techniques by evaluating the retrieval performance on three datasets.

## 1 Introduction

Objects in pattern recognition are frequently represented by means of feature vectors that allow us to imagine objects as belonging to a high dimensional vector space. To compare these representations in most cases the Euclidean distance is used. However, other measures can be considered especially when non-vectorial representations are adopted (e.g. in structural pattern recognition). Our main application domain is in the field of Document Image Retrieval (DIR) whose aim is to identify relevant documents considering image features only (e.g. considering layout-based retrieval or word indexing [1]).

In this paper we focus on the efficient retrieval of objects represented by  $n$ -dimensional points in suitable feature spaces. The framework that we consider is a query by example paradigm: given an  $n$ -dimensional query vector we look for most similar objects by searching the nearest points in the feature space. The simplest implementation relies on an exhaustive comparison of the query with all the indexed vectors, followed by a sorting of the computed distances. It is clear that this approach must be used with caution, for the high computational cost that it implies. To tackle this problem several multidimensional indexing methods have been proposed (e.g. X-tree) to index high-dimensional data more

efficiently than the sequential scan. However, when dealing with “very high dimensional” data (hundreds or thousands of dimensions) many multidimensional indexes degenerate and perform poorly than the sequential scan for reasons that are generally attributed to the so called *curse of dimensionality* [2].

A complementary class of methods adopts a dimensionality reduction of the data as a preliminary processing step, before using a multidimensional index on the reduced space (e.g. [3]). Working on a reduced space the quality of the query results can be reduced, giving rise to wrong results both in terms of false hits and false dismissals ([4] pag 663).

In this paper we describe a dimensionality reduction technique designed to work in the DIR framework previously mentioned. The method is based on the use of Growing Hierarchical Self Organizing Maps (GHSOM) that cluster input vectors into a hierarchy of multiple layers consisting of several independent SOMs. The resulting tree is more deep in presence of more complex clusters. The GHSOM has been mainly used as a visualization tool exploring the maps independently one to each other. In our approach we embed the lower level maps in the root thus obtaining an unique low dimensional space where input patterns are projected by interpolation with respect to the cluster centers.

The basic embedding approach has been described in a recent work [5]. In this paper we focus on an improvement that relies on a non-linear placement of cluster centers in the root map. The Sammon’s algorithm is used to place the neurons of the root map in  $R^2$ . We then compute the Voronoi diagram of the cluster centers to identify the regions in which to embed the second level maps by means of a projective transformation algorithm.

In Section 2 we analyze some methods for dimensionality reduction and we describe the basic characteristics of the Growing Hierarchical Self Organizing Map that are useful to understand the proposed method. In Section 3 we describe our previous approach, the Embedding Map Projection (EMP), and the Nonlinear Embedding Map Projection (NEMP). The experiments are described in Section 4 and some final remarks are drawn in Section 5.

## 2 Related Work

Dimensionality reduction techniques have been studied for a long time resulting in a large collection of methods available. In this section we do not aim at a broad literature survey, rather we summarize the methods that we considered for comparison with our approach.

Principal Components Analysis (PCA) performs dimensionality reduction by embedding the data into a low dimensional space finding a linear basis in which the variance in the data is maximal. PCA is based on the computation of the covariance matrix of the input data followed by the evaluation of the principal eigenvectors of the matrix, that will form the basis of the reduced space. The mapping of points in the reduced space can be computed in a straightforward way with a matrix multiplication. When dealing with real data non-linear mappings are frequently preferred to linear ones (such as PCA). In this framework both global and local techniques can be considered [6].

In the category of global techniques we consider autoencoders that have been used since the 1990's [7]. Autoencoders are Multilayer perceptrons (MLP) having the same number of input and output units and a reduced number of nodes in a hidden layer. During the training the network is forced to reproduce in the output layer the input patterns. The hidden units of a trained network describe the training data with few neurons, performing the desired non-linear dimensionality reduction. Similarly to other MLP-based architectures, large autoencoders can be difficult to train both for the computational cost and for the risk to get stuck in local minima. A new training strategy has been recently proposed [8] allowing large networks to be trained.

Local Tangent Space Analysis (LTSA) is a method that is based on the representation of the local geometry of the manifold using tangent spaces [9]. The local space is estimated by computing the PCA on the  $k$  nearest points of each input point and the local spaces are aligned to obtain the global coordinates of the data points. With LTSA it is not possible to embed additional data points in addition to those used to compute the transformation (*out-of-sample* extension) and it is not possible to index additional objects or to perform queries with objects not indexed.

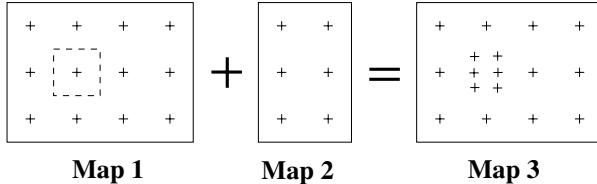
Sammon mapping is a global non linear technique that maps the high-dimensional data representation to a low-dimensional representation while retaining the pairwise distances between the datapoints as much as possible [10]. The Sammon algorithm adjusts the input vectors in the output low-dimensional space trying to minimize a cost function. In Sammon technique, similarly to LTSA, the *out-of-sample* extension is not available. This limit does not affect PCA, autoencoders and the method proposed in this paper.

## 2.1 Growing Hierarchical Self Organizing Map

The Self-Organizing Map (SOM) is a clustering technique that can be considered in the artificial neural networks framework [10]. The SOM neurons are typically arranged in a two dimensional grid and each neuron is associated with a weight vector that corresponds to the cluster centroid. Two main limitations of the basic SOM algorithm are addressed by the GHSOM. First, at the beginning of the training it is required to define the map structure. Second, to accurately represent complex clusters, in some cases we need to build very large maps resulting in computational problems both for the map training and for its use. The GHSOM [11] dynamically models the training data and has been proposed to address the above mentioned problems. The GHSOM allows the network structure to grow in width and in depth, building a data structure with several layers consisting of independent SOMs. During the GHSOM training the map is adapted to the underlying distribution of training patterns.

## 3 Embedded Map Projection

Even if it is not easy to visualize data in high dimensional spaces, several studies have demonstrated that real patterns are unlikely to belong to uniform



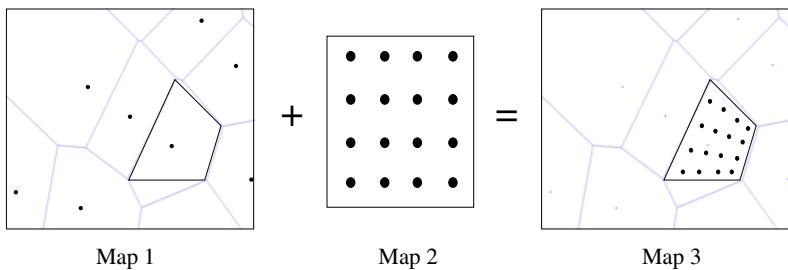
**Fig. 1.** Linear embedding of a second level SOM (Map2) into the parent one (Map1)

distributions in the vector space. The patterns can frequently be imagined as belonging to lower dimensional manifolds or to clusters. For instance, the Cluster Tree [12] has been proposed as an index structure used to perform approximate search in high dimensional spaces on the basis of pattern clustering. In [13] we combined the SOM clustering with the PCA to efficiently index words represented by points in high dimensional spaces. Words in each cluster are projected in a low dimensional space with PCA to speed up the similarity search. In that work we did not use the topological order of clusters since complex patterns, such as words, can not be easily modeled by a single map. One solution is to use a larger SOM, but training such a map is not easy and the retrieval time risks to be very high, since the number of clusters quickly becomes very large. As an alternative approach we proposed in [5] to use the GHSOM hierarchy of maps to define a low dimensional space where input patterns can be projected. To give an idea of the type of maps that are built in the hierarchy, in the upper left part of Figure 3 we report the first level map and two sub-maps computed for the MNIST dataset. The basic idea of this approach is to embed lower maps in an output space that is defined by the first level map, subsequently projecting input points in this space. The GHSOM training, and embedded map building, is performed on a reduced number of points randomly selected from the collection to be indexed. The whole dataset is used in the projection step.

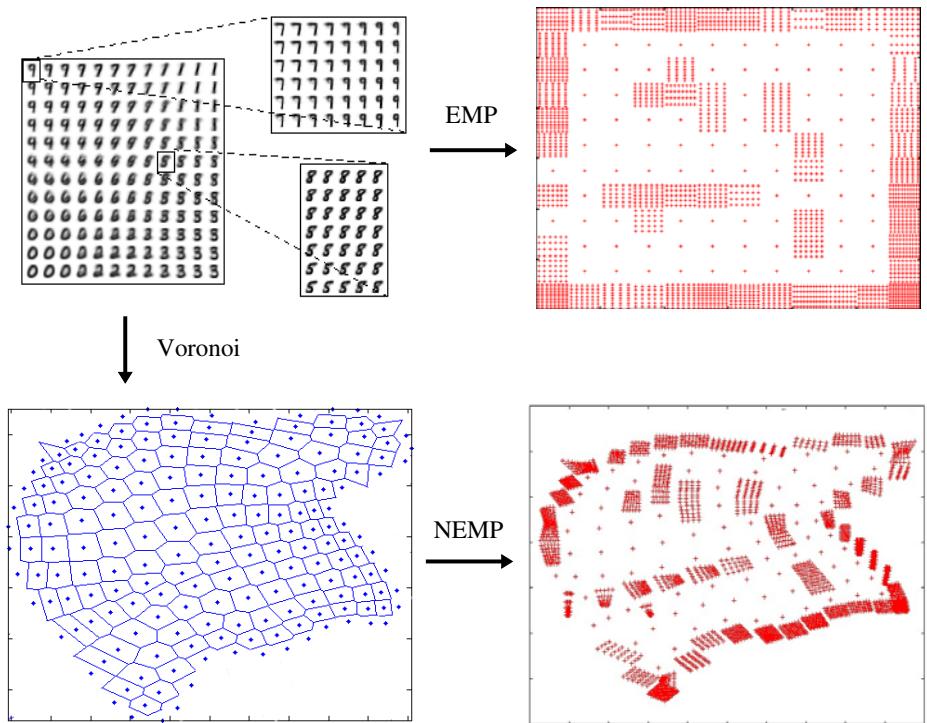
In Figure 1 we show the embedding of a second order map in the corresponding parent map as it is obtained by our previous method. The SOM cluster centers belong to a two-dimensional grid (represented by crosses in Map1). The cluster (1,1) (surrounded by a dashed box) is described with more details by a 3x2 map (Map2). Embedding Map2 into Map1 we obtain Map3 that describes with more resolution the region corresponding to the neuron (1,1) in Map1. The actual embedding is obtained by a recursive linear scaling of lower level maps in the main one. An actual embedded map computed with the MNIST dataset is shown in the upper part of Figure 3.

### 3.1 Nonlinear Embedded Map Building

Figure 2 depicts with one example the non-linear embedding proposed in this paper. After computing the GHSOM representing the input training patterns we re-arrange the neurons of the root level map using the Sammon mapping. The general idea behind this improvement is that a uniformly distributed grid of neurons does not necessarily reflect the similarity of cluster centers. We therefore



**Fig. 2.** Nonlinear embedding of a second level SOM (Map2) into the parent one (Map1)



**Fig. 3.** Embedding obtained for the MNIST dataset. In the upper part we show the root GHSOM and two lower level maps and the linear embedding. The lower part contains the nonlinear embedding.

used the Sammon mapping to move closer clusters in the input space in near positions in the 2-d output space.

This adaptation is obtained by means of three steps that are summarized in the following and exemplified in Figure 2. First, we compute the Sammon mapping of the neurons in the root map. Second, in order to identify the regions

where to embed the second level maps, we compute the Voronoi diagram of the root neurons. At this step, to avoid the open Voronoi polygons problem, before computing the Voronoi regions we add a frame of virtual points computed by an interpolation method (see the bottom-left picture in Fig. 3). Third, after computing the Voronoi diagram, we take into account the root neurons that are expanded in the GHSOM. For each expanded neuron, we find four points ( $P$ ) belonging to the set of vertices of the corresponding Voronoi region. We then embed into  $P$  the neurons of the second level map, a regular grid of neurons, computing a projective homography transformation [14]. In the lower part of Figure 3 we show the embedding obtained with this algorithm in the MNIST dataset. On the left we report the new positions of root neurons after computing the Sammon mapping, together with the Voronoi regions and the virtual points. On the right we show the sub-maps embedded using the homography transformation. The NEMP method is a bit more computationally expensive than EMP method, and the adding computational load is only for the Voronoi computation.

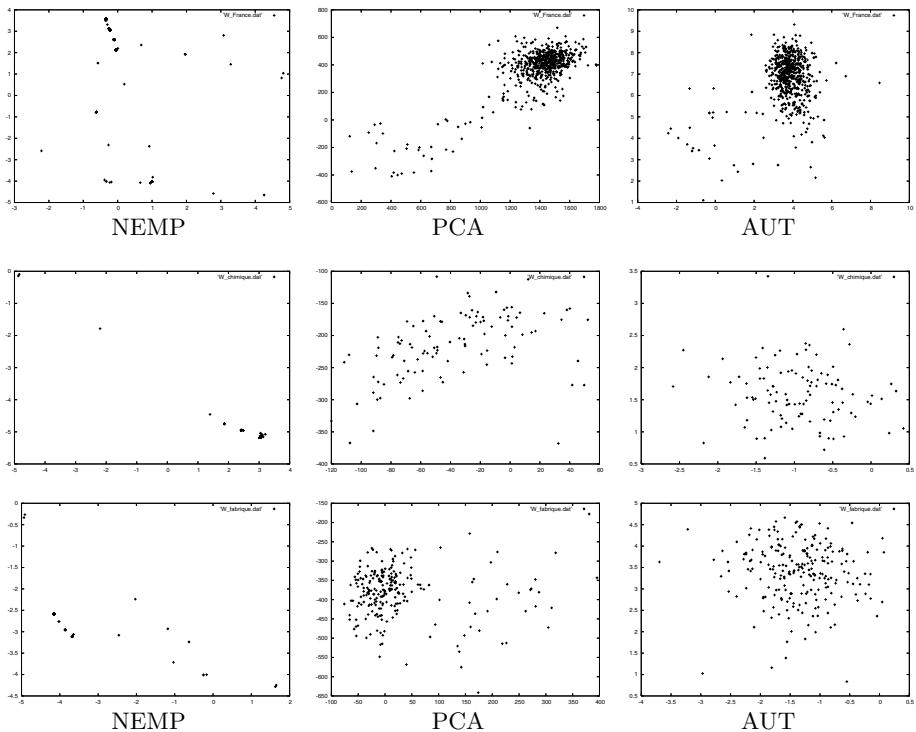
After computing the embedded map we project all the input points in the previously defined two-dimensional space. Previous approaches to perform this projection worked with single SOMs [15], or with each individual map in the GHSOM [16]. The projection described here deals with the embedded map centroids and projects each input point  $x$  as follows (see also [5]). Let  $c_1$  be the closest centroid to point  $x$  and  $c_2$  and  $c_3$  be the next closest centroids with distances  $d_i = \|x - c_i\|$  ( $i = 1, 2, 3$ ). The three distances are ordered so that  $d_1 \leq d_2 \leq d_3$  and  $x$  is placed between the three points according to:

$$x' = \frac{d_1^{-1}c_1 + d_2^{-1}c_2 + d_3^{-1}c_3}{d_1^{-1} + d_2^{-1} + d_3^{-1}}. \quad (1)$$

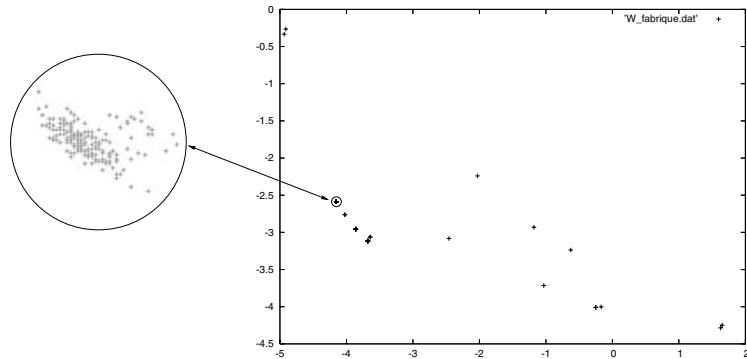
## 4 Experiments

We made several experiments on three datasets comparing some dimensionality reduction methods for similarity retrieval. The PCA, LTSA, and autoencoder software has been described in [17]. The proposed embedded map projection method has been implemented starting from the GHSOM Toolbox for Matlab [11]. In the experiments we used three datasets. The MNIST dataset is a widely used collection of handwritten digits containing 60,000 images. The COIL20 dataset contains images of 20 objects, depicted from 72 different viewpoints each. The WORDS dataset is a collection of digitized printed words normalized to fit a 12 x 57 grid. In this dataset we have 132,956 word images extrapolated from 1,302 pages that are part of an encyclopedia of the *XIX<sup>th</sup>* Century.

We first projected the input data of each dataset into a two dimensional space using the compared projection methods. Details of the parameters used for PCA, LTSA and autoencoders can be found in [5]. For the GHSOM training we evaluated several combinations of parameters, but for all the datasets the best results have been achieved with  $\tau_1 = 0.6$  and  $\tau_2 = 0.005$ . The resulting maps have



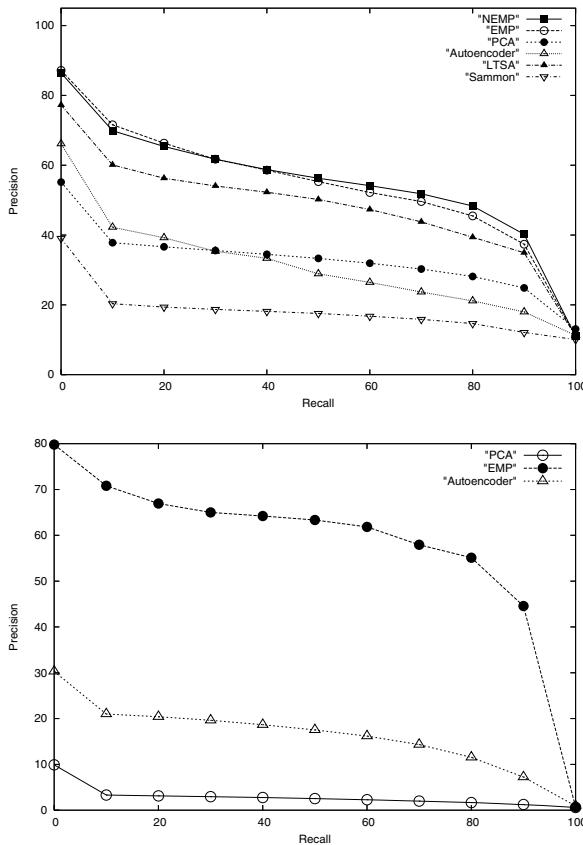
**Fig. 4.** Projection of the WORDS dataset for three words (*France*, *chimique*, and *fabrique*)



**Fig. 5.** Detail of the NEMP for the second word in Figure 4

**Table 1.** Precision at 0 percent Recall with the four compared methods

Dataset	$n$	Train	Indexed	NEMP	EMP	PCA	Autoencoder	Sammon	LTSAs
MNIST	784	10,000	10,000	86.44	87.14	55.17	66.14	39.06	77.25
		10,000	60,000	84.00	84.56	53.55	64.66	—	—
COIL20	1024	1,440	1,440	86.59	85.59	71.68	82.35	63.58	43.68
WORDS	684	13,296	132,956	86.60	84.54	12.55	23.03	—	—

**Fig. 6.** Precision-Recall plots for the compared methods working on the MNIST (top) and WORDS (bottom) data-sets

the following features<sup>1</sup>: MNIST(2,60,2566), COIL20(4,70,395), WORDS(2,60, 2077). For the Sammon algorithm the iterative process for the minimization of the cost function was made with 100 epochs for all datasets.

---

<sup>1</sup> The notation used is DATASET(# levels, # maps, # clusters).

Figure 4 shows the projections of all the occurrences of three words (585 for *France*, 108 for *chimique*, 232 for *fabrique*) in the WORDS dataset with the proposed method (NEMP), PCA and the autoencoders. The NEMP method arranges the points of the words in narrow areas in the output space as it can be verified also by the detail in Figure 5. This feature is particularly useful for the use of dimensionality reduction to perform similarity search.

To obtain a numerical evaluation we measured the accuracy achieved by a query by example retrieval performed on the reduced space. We made several queries and we computed a Precision-Recall plot averaging each query. For the COIL20 dataset we used in turn each point as query evaluating the retrieval performance. For MNIST we used 10,000 queries randomly selected from the whole 60,000 patterns. In the case of the WORDS dataset we used the 111 most frequent words with a total of 24,050 occurrences. Also in this case we used in turn each word as a query with a significantly higher number of queries with respect to the experiments described in [5] (where we considered only 576 words).

Figure 6 shows the Precision-Recall plots for the MNIST and WORDS datasets. In the MNIST dataset we projected only the 10,000 training patterns, and therefore we have also a plot for LTSA and Sammon. In the WORDS dataset the LTSA and Sammon plots are missing, since we indexed all the 132,956 points, and as mentioned in section 2, the out-of-sample extension is not available for these methods. We summarize in Table 1 all the performed experiments reporting the Precision at Recall 0 in various cases (this value is obtained by an interpolation of the Precision Recall plots). In the table,  $n$  is the size of the input space. We report also the number of objects used for training (*Train*) and the number of indexed objects (*Indexed*). Summarizing, it is clear that the method described in this paper outperforms the other approaches and also the EMP one, in particular with the WORDS dataset. The increase in the computational cost with respect to the EMP is limited to the computation of the Voronoi diagram.

## 5 Conclusions

In this paper we propose a dimensionality reduction method that is based on the embedding of lower level maps of a GHSOM clustering of the input data. The method has been compared with other dimensionality reduction methods on a query by example retrieval application on three datasets. These preliminary results are encouraging, since the NEMP method outperforms the compared ones. Current work is addressing the extension of the approach to reduced spaces of size greater than two, in order to expand the possible range of applications.

## References

1. Marinai, S., Marino, E., Soda, G.: Font adaptive word indexing of modern printed documents. *IEEE Transactions on PAMI* 28(8), 1187–1199 (2006)
2. Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning; data mining, inference, and prediction*. Springer series in statistics, New York (2001)

3. Kanth, K.V.R., Agrawal, D., Singh, A.: Dimensionality reduction for similarity searching in dynamic databases. *SIGMOD Rec.* 27(2), 166–176 (1998)
4. Samet, H.: Foundations of multidimensional and metric data structures. Morgan Kaufmann, Amsterdam (2006)
5. Marinai, S., Marino, E., Soda, G.: Embedded map projection for dimensionality reduction-based similarity search. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) *S+SSPR 2008*. LNCS, vol. 5342, pp. 582–591. Springer, Heidelberg (2008)
6. van der Maaten, L., Postma, E., van den Herik, H.: Dimension reduction: A comparative review (preprint, 2007)
7. DeMers, D., Cottrell, G.: Nonlinear dimensionality reduction. In: *NIPS-5* (1993)
8. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* 313(5786), 504–507 (2006)
9. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM Journal of Scientific Computing* 26(1), 313–338 (2004)
10. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection. *IEEE Transactions on Neural Networks* 11(3), 574–585 (2000)
11. Chan, A., Pampalk, E.: Growing hierarchical self organising map (ghsom) toolbox: visualisations and enhancements. In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP 2002*, vol. 5, pp. 2537–2541 (2002)
12. Li, C., Chang, E., Garcia-Molina, H., Wiederhold, G.: Clustering for approximate similarity search in high-dimensional spaces. *IEEE Transactions on Knowledge and Data Engineering* 14(4), 792–808 (2002)
13. Marinai, S., Faini, S., Marino, E., Soda, G.: Efficient word retrieval by means of SOM clustering and PCA. In: Bunke, H., Spitz, A.L. (eds.) *DAS 2006*. LNCS, vol. 3872, pp. 336–347. Springer, Heidelberg (2006)
14. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2003)
15. Wu, Z., Yen, G.: A som projection technique with the growing structure for visualizing high-dimensional data. In: *Proceedings of the International Joint Conference on Neural Networks, 2003*, vol. 3, pp. 1763–1768 (2003)
16. Yen, G.G., Wu, Z.: Ranked centroid projection: a data visualization approach with self-organizing maps. *IEEE Transactions on Neural Networks* 19(2), 245–258 (2008)
17. van der Maaten, L.: An introduction to dimensionality reduction using matlab. Technical Report Technical Report MICC 07-07 (2007)