Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 57 (2004) 49–76
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis

John Aldo Lee[a,*], Amaury Lendasse[b], Michel Verleysen[a,1]

[a] *Microelectronics Laboratory, Department of Electricity, Université catholique de Louvain, Place du Levant, 3, B-1348 Louvain-la-Neuve, Belgium*

[b] *Université catholique de Louvain, CESAME, Avenue George Lemaître, 4, B-1348 Louvain-la-Neuve, Belgium*

Abstract

Dimension reduction techniques are widely used for the analysis and visualization of complex sets of data. This paper compares two recently published methods for nonlinear projection: Isomap and Curvilinear Distance Analysis (CDA). Contrarily to the traditional linear PCA, these methods work like multidimensional scaling, by reproducing in the projection space the pairwise distances measured in the data space. However, they differ from the classical linear MDS by the metrics they use and by the way they build the mapping (algebraic or neural). While Isomap relies directly on the traditional MDS, CDA is based on a nonlinear variant of MDS, called Curvilinear Component Analysis (CCA). Although Isomap and CDA share the same metric, the comparison highlights their respective strengths and weaknesses.

Keywords: Nonlinear projection; Nonlinear dimensionality reduction; Geodesic distance; Curvilinear distance

1. Introduction

When analyzing huge sets of numerical data, problems often occur when the raw data are high-dimensional. For example, these difficulties are typical in domains like image processing (large number of pixels) or biomedical signal analysis (numerous captors).

* Corresponding author. Tel.: +32-2-47-81-33; fax: +32-2-47-25-98.
E-mail addresses: lee@dice.ucl.ac.be (J.A. Lee), lendasse@auto.ucl.ac.be (A. Lendasse), verleysen@dice.ucl.ac.be (M. Verleysen).
[1] M.V. works as a senior research associate of the Belgian FNRS.

0925-2312/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2004.01.007

From a theoretical point of view, two related facts explain the problems encountered in high-dimensional spaces:

- The *curse of dimensionality* [1]. The number of samples required to approximate accurately a data distribution grows exponentially with the dimensionality.
- The *empty space phenomenon* [19]. Some properties of high-dimensional spaces are unexpected (for example, the volume of a sphere inscribed in a cube tends to zero when the dimensionality grows).

These phenomenons have insidious consequences, sometimes leading to malfunctions in the analysis algorithms. A solution is to reduce the dimension of the raw data, because their structure is often more simple than they look at first sight, i.e. they probably contain redundancies and dependencies between the variables. This means that the raw data often lie on a manifold whose dimension is smaller than the dimension of the embedding space. Under this assumption, the dimension reduction is achieved by the construction of a continuous mapping between the embedding space and the unknown manifold space. To be useful, this mapping has to be inversible (in the noiseless case), in order to project and reconstruct the data with minimal error.

An essential parameter to build the mapping is the intrinsic dimension [6,7] of the data, i.e. the true dimensionality of the manifold, given by its number of latent (or explicative) variables. Most dimension reduction techniques require to know the intrinsic dimension in advance, in order to avoid both underfitting or overfitting.

From a practical point of view, the dimension reduction techniques can be classified according to

- the class of model,
- the criterion they optimize and,
- the type of application for which they are designed (e.g. visualization).

First, linear and nonlinear models can be distinguished. Among the linear ones, three well-known methods are Principal Component Analysis [9,12], Projection Pursuit [5,10] and the original metric multidimensional scaling [21]. The PCA criterion yields a linear projection which preserves as much as possible the variance of the manifold. Practically, this is achieved by computing the eigenvectors of the covariance matrix of the data. Projection Pursuit generalizes the PCA by allowing the user to select a criterion more sophisticated than the decorrelation. For example, statistical independence is the criterion which leads to Independent Component Analysis [11]. Finally, the metric MDS works like the PCA, by extracting eigenvectors (in this case from the matrix of pairwise distances).

Among the nonlinear models, a first one may be local PCA [13]. After a vector quantization of the data, several PCA are conducted, one on each cluster. This locally linear model is nonlinear at the level of the whole manifold. But unfortunately local PCA is not designed to represent the manifold in a single coordinate system. This obviously compromises any attempt to consider the data set as a whole for any subsequent continuous process.

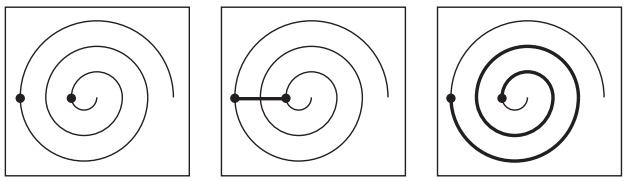

Fig. 1. Distance between two points: (a) two points on a manifold with a spiral shape, (b) the Euclidean distance between them in the data space, (c) the distance along the manifold, which is the one approximated by the curvilinear or geodesic distance.

Nonlinear methods not suffering from this disadvantage are nonlinear variants of the metric MDS, like Sammon's nonlinear mapping (NLM) [18] or the Curvilinear Component Analysis (CCA) [3,8]. These algorithms minimize criterions which completely differ from the one used by PCA. They are based on notions like topology and neighborhood. Actually, they build a mapping in such a way that the pairwise distances between the raw data vectors are reproduced between the mapped vectors. These algorithms show good capabilities for the unfolding of nonlinear manifolds. Their limitations come from the distortions that can exist between the distances measured in the data space and the distances measured in the manifold space (see the example of the spiral in Fig. 1). The last evolutions of the distance-preserving techniques avoid partly the problem by using a more complex metric than the Euclidean distance. For example, the Curvilinear Distances Analysis (CDA) [15,16] and Isomap [20], developed independently and compared in this paper, compute a 'curvilinear' or 'geodesic' distance [2]. This metric can measure good approximations of the distances along the manifold, without shortcuts as does the Euclidean distance (Figs. 1b and c). By the way, it is worth to notice that the geodesic distance could be used not only in MDS and CCA, but also in Sammon's NLM. However, such a combination has apparently not been published yet and is not investigated here.

The remainder of this document is organized as follows. Section 2 explains and defines the curvilinear (or geodesic) distance. Sections 3 and 4, respectively, detail how Isomap and CDA works. In order to compare both algorithms, Section 5 shows some experimental results and Section 6 discusses them. Finally, Section 7 draws the conclusions and sketches some perspectives for future work.

2. Curvilinear distance

At the first glance, the curvilinear distance appears as a very strange concept. Indeed, although it is directly apparented with the Euclidean distance, the curvilinear distance depends not only on the two points between which the distance is measured, but also on other surrounding points. This use of more than two points is totally unknown in the world of L_p norms, where e.g. Euclidean and Manhattan distances are coming from. A well-known exception is the Mahalannobis distance, for which the covariance

of the data set is taken into account. Nevertheless, this approach remains very limited by comparison with the curvilinear distance. Actually, the curvilinear distance does not use global statistics computed on the data set, but it exploits the shape of the data set at a local level.

Intuitively, the goal of the curvilinear distance consists in computing distances along an object. For example, a plane cannot fly from New York to Tokyo by following a straight line (a plane is neither a submarine nor a tunneller!). Instead, it has to follow the curvature of the Earth. This comparison explains why curvilinear distances are also known as geodesic distances. Of course, the curvilinear distances must not be restricted to the computation of curves on the surface of a sphere. It has to be feasible for any manifold, and especially for manifolds that are only known through set of points.

Since only a discrete representation of the manifold is known, only a discrete approximation of the curvilinear distance will be computed. Instead of measuring the length of a curve, we could sum the lengths of small interconnecting segments that approximate the curve. Two important questions remain. Starting with the points in the data set, how to build the segments? And how to choose the interconnecting segments that best approximate the true curvilinear distances?

The first question has a direct answer. If two points are neighbors, i.e. one of the two points is the closest one to the other, then it seems normal that a segment should connect the two points; the curvilinear distance between them is simply the length of the segment. This reasoning gives a clue about how to weave the links between the points. Obviously, not only the closest point may be used to build a link to a point: in high-dimensional space, a point can be surrounded by several other ones. Therefore, a point would ideally be connected to the k nearest ones (k-rule), or with all points lying closer than a certain threshold ε (ε-rule). To the end, if every point is connected with all other ones, the curvilinear distance is equivalent to the norm used to measure the length of the segments.

The second question does not find an answer so easily. Fortunately, the data set is now complemented with links and can be seen as a weighted graph structure. Indeed, each link can be labelled with its (Euclidean) length. Such a structure has been widely studied in computer science and graph theory. These fields abound in algorithms intended to compute minimal cost trajectories in graphs. Especially, Dijkstra's algorithm [4] computes shortest paths in a weighted graph. Given a point in a graph, Dijkstra's algorithm computes the length of the shortest path to each other (reachable) point by summing the length of the segments along the shortest path. Under certain conditions on the links, it can be proven [2] that the length computed by Dijkstra's algorithm tends to the true curvilinear distance when the number of points defining the manifold increases. In this case, indeed, the length of the segments tends to decrease and the shortest paths between pairs of points become smoother and smoother.

In the framework of nonlinear projection by distance preservation, the purpose of the curvilinear distance consists in simplifying the preservation of long distances when dealing with a nonlinear manifold. Indeed, the example of the spiral of Fig. 1 is intuitively clear: the correct unfolding of the spiral requires to unroll it. However, when doing so, long Euclidean distances have to be stretched, contrary to curvilinear ones, which are naturally longer.

3. Isomap

Isomap [20] is the straightest way to use the curvilinear distance for nonlinear projection. In this section, the term curvilinear is replaced by geodesic, since it is the word used by the authors of [20].

First of all, assume that all n data vectors $x_{1\leqslant i\leqslant n}$ are stored in matrix X as d-dimensional columns: $X=[x_1,\ldots,x_i,\ldots,x_n]$.

Classically, e.g. in psychology, when high-dimensional sets of points are only known by their pairwise distances (or equivalently by their similarity), the traditional PCA cannot be achieved (the coordinates X of the points are not known). But, assuming that the distance or similarity measures are convertible to Euclidean distances, the equivalence between PCA and traditional metric MDS can be exploited in order to find the same solution as PCA would have given. Instead of finding the principal components as eigenvectors of XX^{T} (proportional to the sample estimation of the covariance matrix of the set X of unknown coordinates), the dual problem $X^{\mathrm{T}}X$ can be investigated. Actually, $X^{\mathrm{T}}X$ relates to D, the $n\times n$ matrix of all pairwise distances. In this case, the eigenvectors of $X^{\mathrm{T}}X$ are directly proportional to the coordinates along the principal components. The main shortcoming of both PCA and MDS is that they can only deal with linear dependencies between the features.

In this framework, Isomap replaces the Euclidean distances by the geodesic ones. This modification makes the projection of nonlinear manifolds possible, at least for a certain class of manifolds. This class gathers all compact smooth submanifolds of $\mathbb{R}^d$ (called intrinsically Euclidean manifolds in [2]) that can be isometrically mapped to a convex domain of $\mathbb{R}^p$. Here, the term 'isometrically' means that the mapping equates the Euclidean distances in $\mathbb{R}^p$ with the true manifold distances measured according to the natural Riemannian structure of the manifold and induced from the Euclidean metrics on $\mathbb{R}^d$. Intuitively, considering only 2D manifolds, all manifolds obtainable by curving a sheet of paper belong to class of Euclidean manifolds. More generally, a Euclidean manifold must respect the following scheme:

$$\begin{bmatrix} x_{(1)} \\ \vdots \\ x_{(r)} \\ \vdots \\ x_{(d)} \end{bmatrix} = A \begin{bmatrix} f_{(1)}(y_{(1\leqslant s\leqslant p)}) \\ \vdots \\ f_{(r)}(y_{(1\leqslant s\leqslant p)}) \\ \vdots \\ f_{(d)}(y_{(1\leqslant s\leqslant p)}) \end{bmatrix}, \tag{1}$$

where A is a square matrix and the bracketted indices $(1\leqslant r\leqslant d)$ and $(1\leqslant s\leqslant p)$, respectively, indicates the rth and sth feature of the vector x (data vector in embedding space) and y (corresponding vector in parameter space). Functions $f_{(1\leqslant r\leqslant d)}$ are one-to-one.

Practically, Isomap goes through the following steps:

(1) Select randomly l points ($l < n$): when the data set X contains numerous points, it is useful to select a random subset P of them when measuring the pairwise distances in order to keep the number of pairs computationally tractable.
(2) Connect neighboring points: connect each point either with the k closest other ones (k-rule), or with those lying closer than a certain threshold ε (ε-rule).
(3) Compute the matrix D of all pairwise geodesic distances: run Dijkstra's algorithm for each point and store the pairwise distances between all points in a symmetric matrix D with $l \times l$ entries.[2]
(4) Center D: compute the mean of the rows, the mean of the columns and the mean all entries; subtract the mean of the rows from each row, subtract the mean of the columns from each column and the grand mean to all entries; this makes D equivalent to $P^{\mathrm{T}}P$.
(5) Compute the eigenvalues and eigenvectors of the centered D and sort eigenvectors according to the descending order of their associated eigenvalues.
(6) Choose p such that the residual variance associated with the $l-p$ last eigenvectors is sufficiently small; suppress those eigenvectors.

The p kept eigenvectors give the coordinates $M = [m_1, \ldots, m_i, \ldots, m_l]$ of the mapped points in a p-dimensional projection space.

4. Curvilinear distance analysis

In the same way Isomap is derivated from classical metric MDS, Curvilinear Distance Analysis (CDA, see [15,16]) comes from Curvilinear Component Analysis (CCA, see [3,8]), intended to be a nonlinear alternative to PCA. As well CDA as CCA may be seen as neural version of Sammon's nonlinear mapping (NLM, see [18]), conceived as a nonlinear version of metric MDS using gradient descent instead of eigenvalue decomposition.

Similarly to Isomap, CCA/CDA begins by selecting a subset of the data in order to keep the number of pairwise distances computationally tractable. But contrarily to Isomap, the coordinates of the subset P are not randomly drawn in the data set X; instead, they result from a vector quantization applied on the data. Any basic vector quantization can fit; concretely, the implementation uses hard competitive learning (on-line algorithm, decreasing learning rate). Although vector quantization may fall in local minima, the resulting prototypes are far more representative of the original manifold than randomly selected points. As for Isomap, the quantization step may be skipped when the number of data vector is small.

[2] In addition to the selection of a randomly chosen subset of the points in the database, Isomap can also decrease the size of D by making it sparse: all pairwise distances are not computed; instead, only a few points serve as landmarks and the distance from all points to the landmarks are measured. This possibility is not investigated here.

The second stage of CCA/CDA is the mapping procedure itself. While Isomap relies on algebraical methods resulting from the reformulation of the PCA problem as a distance preservation problem (metric MDS), CCA/CDA considers directly this last point of view. Concretely, it means that CCA/CDA works by optimizing a criterion that explicitly measures the preservation of the pairwise distances:

$$E_{\mathrm{CCA/CDA}} = \sum_{\substack{1 \leqslant i \leqslant l \\ i < j \leqslant l}} (\delta_{i,j} - d_{i,j})^2 F(d_{i,j}, \lambda), \tag{2}$$

where $\delta_{i,j}$ is a distance measured between prototypes p_i and p_j in the data space and $d_{i,j}$ is a distance measured between the coordinates m_i and m_j of the same two prototypes in the projection space. The factor $F(d_{i,j}, \lambda)$ weighs the contribution of each pair of prototypes in the criterion. Contrarily to the NLM, the factor F does not depend on the distance $\delta_{i,j}$ in the data space but on the distance $d_{i,j}$ in the projection space, like for a Self-Organizing Map (SOM, see [14,17]). Consequences of this important change will be analyzed later on. Moreover, the factor F does not need to be static: it may be a function that evolves during the convergence by means of the parameter λ, again like in a SOM. This parameter may be seen as a neighborhood radius controlling the scale at which the unfolding of the manifold occurs. Usually, F is implemented as the Heaviside step function:

$$F(d_{i,j}, \lambda) = \theta(\lambda - d_{i,j}) = \begin{cases} 0 & \text{if } \lambda - d_{i,j} < 0, \\ 1 & \text{if } \lambda - d_{i,j} \geqslant 0. \end{cases} \tag{3}$$

Starting from the criterion, the derivation of the learning rule follows a similar scheme as for a stochastic gradient descent. Nevertheless, instead of moving one mapped prototype according to the position of all other ones, one prototype m_i is frozen while moving all other ones ($m_{j \neq i}$) radially around it:

$$m_j \leftarrow m_j + \alpha F(d_{i,j}, \lambda)(\delta_{i,j} - d_{i,j}) \frac{m_j - m_i}{d_{i,j}}, \tag{4}$$

where α and λ are, respectively, a time-decreasing learning rate and the neighborhood radius. More details about the choice of F and the derivation of the learning rule can be found in [3].

Actually, the main difference between CCA and CDA regards the metric $\delta_{i,j}$ which measures the distance between two prototypes. While CCA uses the traditional Euclidean distance, CDA works with the curvilinear distance like Isomap. Although CCA is more powerful than NLM, its main shortcoming is the parameterization of the neighborhood radius λ. When the manifold is nonlinear, λ has to be large enough to get a not too slow unfolding, but it also has to be small enough to avoid the preservation of long Euclidean distances which have to be distorted anyway. In this situation, the use of the curvilinear distance simplifies the parameterization since the entire class of Euclidean manifolds can be unfolded without care of λ. Its value has simply to be set beyond the maximum curvilinear distance measured in the data set.

By comparison with CCA, CDA also reinforces its neural aspect. Indeed, as in CCA, the prototypes coming from the vector quantization are considered as neurons, i.e. processing units independent from the learning set. These neurons have connections

towards both the data and projection spaces. But in addition, the use of the curvilinear distance, and particularly its implementation as a proximity graph, may be seen as a way to weave lateral connections. In that sense, CDA shows more or less the same structure as a SOM. Another resemblence regards the two learning parameters α and λ, which have approximately the same meaning in both methods.

5. Comparison between CDA and Isomap

This section highlights some differences between the results of Isomap and CDA. The first subsection gives an intuition of how both methods work, with the help of toy experiments. The second subsection, although still based on artificial data, affords real-life problems in the field of image processing.

5.1. Toy experiments

In order to explain visually and intuitively the capabilities of Isomap and CDA, nothing compares to two-dimensional manifolds embedded in a three-dimensional space. Fig. 2 illustrates four examples. All of them are intrinsically two-dimensional and can thus be projected on a plane.

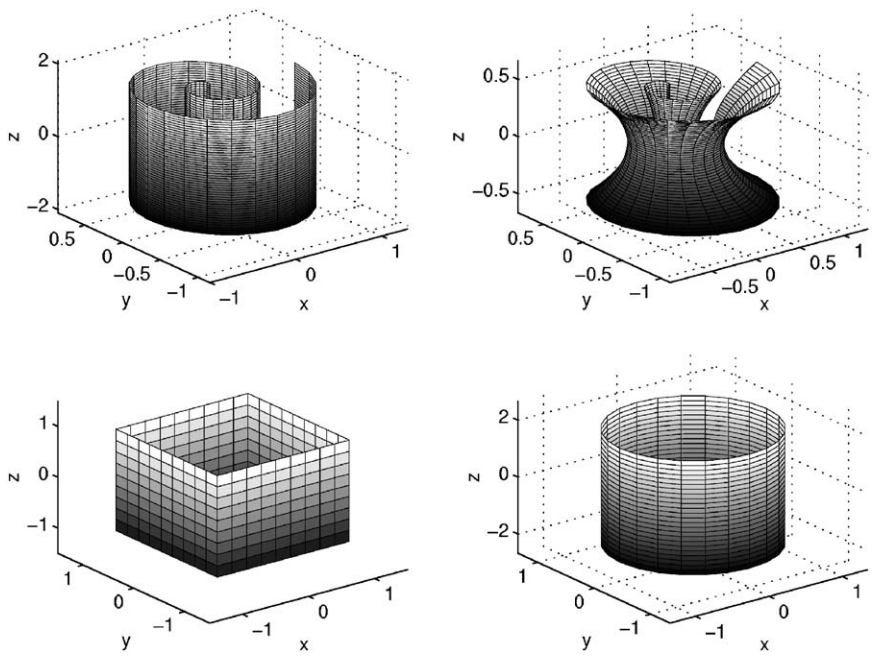

Fig. 2. Four typical manifolds: the Swiss roll (top left), the heated Swiss roll (top right), the open box (bottom left) and the cylinder (bottom right).

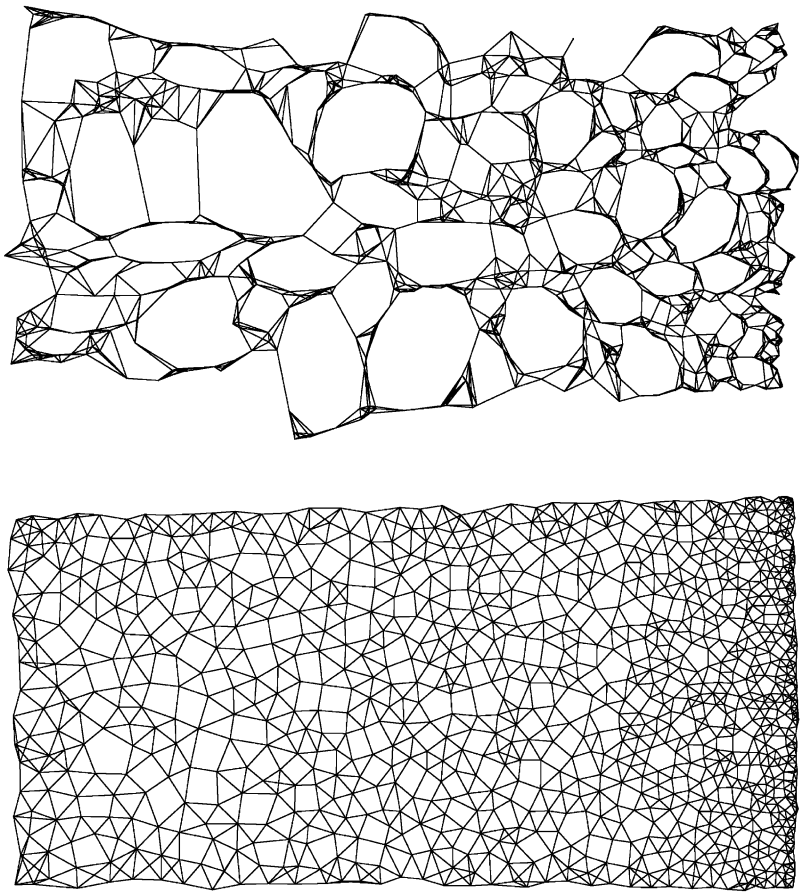

Fig. 3. Two-dimensional projections of 20,000 samples drawn from the Swiss roll manifold by Isomap with random selection (top) and by CDA (bottom); the number of randomly selected points or prototypes is 1000 in both cases; $k = 5$.

The first one (see Fig. 2 top left) is the well-known Swiss-roll, more or less similar to the one already used in [20] to explain Isomap. Its parametric equations are

$$x = u\cos(4\pi v), \tag{5}$$

$$y = u\sin(4\pi v), \tag{6}$$

$$z = \pi(0.5 - u), \tag{7}$$

where u and v vary between 0 and 1. The Swiss-roll belongs to the class of Euclidean manifolds since it suffices to curve a sheet of paper to get it. Alternatively, it can be seen that the above parametric equations match the pattern proposed in Eq. (1). For the experiment, the data set given to Isomap and CDA contains 20,000 points

Table 1
Residual variances given by Isomap for all examples: the size of the projected subset and the way it is chosen (either by Random Selection or Vector Quantization) are indicated in the second and third columns; the fourth column shows the parameter k (minimal number of neighbors for the graph construction); the remaining columns show the residues for projection dimensionalities ranging from 1 to 6. The ideal dimensionality according to Isomap is given by the first column of the plateau where residues hardly vary around a small value; the true intrinsic dimensionality of all manifolds is two; contrarily to eigenvalues yielded by PCA or MDS, residual variances may not decrease strictly

Manifold	Size	Selection	k	1D	2D	3D	4D	5D	6D
Swiss roll	1000	RS	5	0.1081	*0.0042*	0.0030	0.0029	0.0030	0.0029
	1000	VQ	5	0.1054	*0.0007*	0.0010	0.0010	0.0011	0.0011
	1000	VQ	6	0.1783	0.0353	*0.0132*	0.0173	0.0202	0.0207
Heated Swiss roll	1000	VQ	6	0.0341	0.0184	*0.0082*	0.0031	0.0022	0.0014
Box	500	VQ	6	0.6349	0.1702	*0.0356*	0.0433	0.0502	0.0498
Cylinder	500	VQ	6	0.5391	0.3057	*0.0295*	0.0334	0.0352	0.0357
Face	698	no	6	0.197	0.089	*0.023*	0.018	0.014	0.013
Clock	720	no	2	0.570	*0.042*	0.049	0.053	0.054	0.054
Thin Swiss roll	1000	VQ	6	*0.00095*	0.00087	0.00077	0.00086	0.00089	0.00095

drawn from the manifold. After selecting 1000 points and connecting each of them with their five nearest neighbors (k-rule, $k = 5$), Isomap delivers the result shown at the top of Fig. 3, with residue values given in the first row of Table 1. After quantizing and connecting with the same numbers of prototypes and neighbors, CDA yields the result shown at the bottom of Fig. 3. This first example proves the benefit of using the curvilinear or geodesic distance: as well Isomap as CDA unroll the manifold with ease (without any tuning of the α and λ parameters in the case of CDA). However, the results are not completely identical. Indeed, the projection of the Swiss-roll done by Isomap looks like a slice of Swiss cheese, with holes and bubbles, although the probability distribution of the data was not so chaotic before the projection. This effect is due to the random selection of points for Isomap and implies more than only visual consequences on the result. Actually, the random selection does not reproduce correctly the initial distribution of the data set. Thus, some regions are empty whereas other ones are slightly too dense by comparison. As a first consequence, those holes prevent a good approximation of the geodesic distances, that are unnecessarily stretched (they have to slalom around the holes). Secondly, Isomap aggregates points that lie close to each other and widens the holes. In fact, Isomap behaves in that way because it tends to better preserve long distances by comparison with shorter ones, that are shrunken.

Of course, nothing prevents the use of vector quantization as a preprocessing for Isomap as it is done for CDA (see Fig. 4). Indeed, the above described phenomenum does not appear in the result of CDA, for which the vector quantization gives prototypes that are far more representative of the initial distribution in the manifold. From now on, Isomap uses vector quantization in all subsequent examples.

The same manifold can be projected again by Isomap and CDA with vector quantization for both methods. The sole difference regards the number of neighbors, which is increased ($k = 6$). This change causes the apparition of a parasitic link, as shown in the three-dimensional view of Fig. 5. The results of Isomap and CDA are

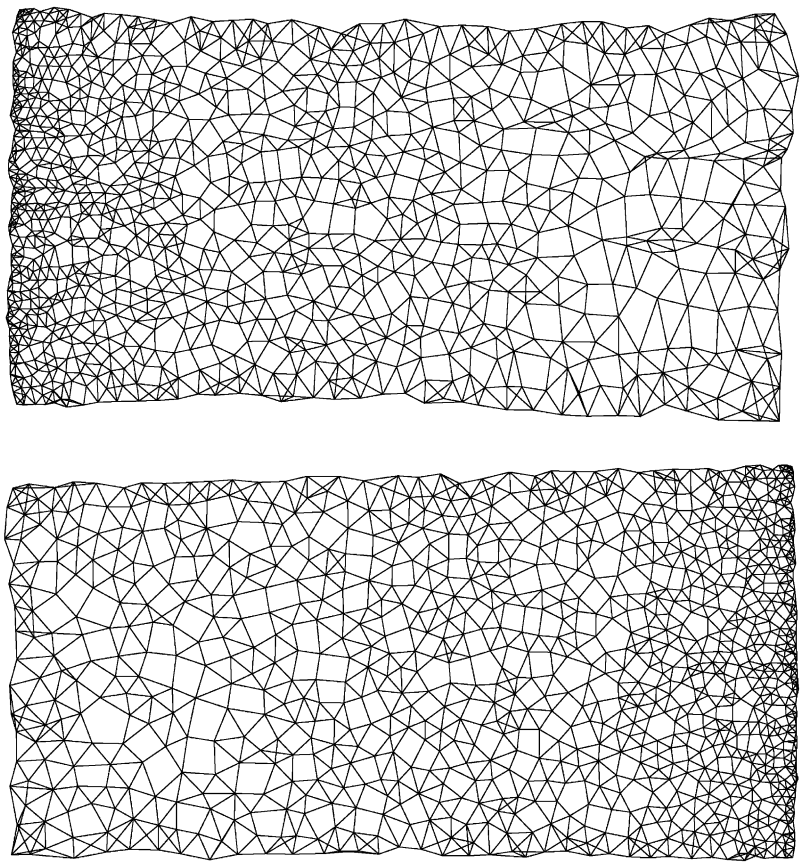

Fig. 4. Two-dimensional projections of 20,000 samples drawn from the Swiss roll manifold by Isomap (top) and CDA (bottom) with vector quantization (1000 prototypes) in both cases; $k = 5$.

illustrated in Fig. 6. The residual variances, in the third row of Table 1, indicate that Isomap difficultly compute a two-dimensional projection. According to the variances, Isomap cannot reduce the dimensionality, just because of the parasitic connection. On the other hand, CDA seems to be more robust: the parasitic link has been stretched (it can be seen on the top right corner of the projection).

The second example (see Fig. 2 top right) is very similar to the Swiss roll. Visually, the original Swiss roll has been slightly heated and molted. The wall follows a parabolic curve instead of a straight line. The number of available points remains 20,000, generated by the parametric equations

$$x = 0.5(1 + (1 - 2r2)^2)u\cos(4\pi v), \tag{8}$$

$$y = 0.5(1 + (1 - 2r2)^2)u\sin(4\pi v), \tag{9}$$

$$z = \pi(0.5 - u), \tag{10}$$

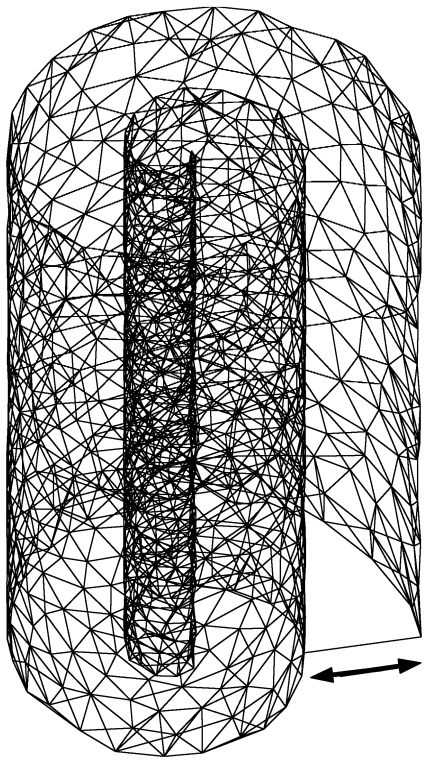

Fig. 5. Three-dimensional view of the 1000 prototypes after vector quantization on 20,000 samples drawn from the Swiss roll manifold; when $k = 6$, a parasitic link appears.

where u and v vary between 0 and 1. Unfortunately, the multiplication by the parabolic factor implies that the manifold is not Euclidean anymore: the above parametric equations does not respect the scheme of Eq. (1). As a consequence, Isomap (1000 selected points, $k = 6$, top of Fig. 7) has some difficulties to find a 2D representation. The residual variances, shown in Table 1 confirm this fact: the plateau of small variances starts at best at the third value. Anyway, the absence of a clear fall between the first and second variances indicates that two dimensions do not suffice to build a satisfying two-dimensional projection with Isomap, although the manifold is intrinsically two-dimensional. On the contrary, CDA (1000 prototypes, $k = 6$, bottom of Fig. 7) benefits from its gradient descent method, allowing some distortions for the large distances.

The third example is an open box (see Fig. 2 bottom left), made of five joined square planes, with a uniform distribution between -1 and $+1$. Clearly, the open box does not belong to the class of Euclidean manifolds. Like the second example, the open box tries to highlight the differences between Isomap and CDA in the mapping phase. The results of Isomap and CDA are shown in Fig. 8, again with 20,000 initial points

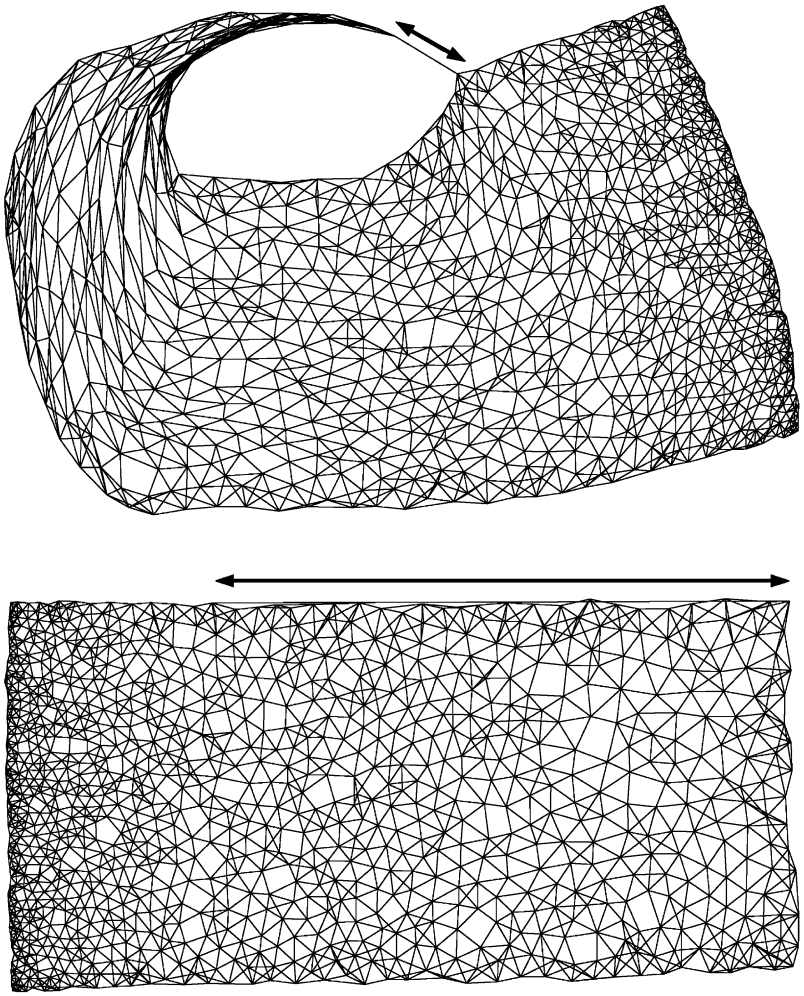

Fig. 6. Two-dimensional projections of 20,000 samples drawn from the Swiss roll manifold by Isomap (top) and CDA (bottom) with vector quantization (1000 prototypes) in both cases; $k = 6$.

but only 500 prototypes (vector quantization for both methods). Visually, Isomap has rounded the sharp edges of the box and shrunken the four lateral faces like a crown: the height of the box is lost. On the other hand, CDA has unfolded the box by tearing two faces (see the stretched links), but the rest of the surface is perfectly flattened and never crushed. This is the consequence of the gradient descent technique used by CDA: the neighborhood factor F allows the algorithm to converge locally. More precisely, the dependence of F on the distance $d_{i,j}$ in the projection space and not on the distance $\delta_{i,j}$ in the data space allows CDA to tolerate stretched links. Indeed, a stretched link implies a longer $d_{i,j}$, and therefore a weaker contribution in $E_{\mathrm{CCA/CDA}}$.

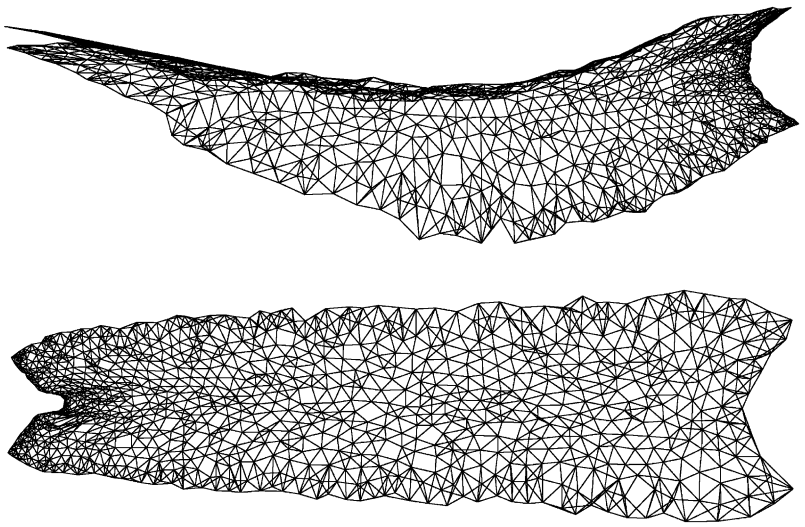

Fig. 7. Two-dimensional projections of 20,000 samples drawn from the heated Swiss roll manifold by Isomap (top) and CDA (bottom); the 1000 shown points are the prototypes determined by CDA, which are directly fed into Isomap instead to select 1000 points randomly.

Isomap, on the other hand, can only deliver a global linear solution downstream from the nonlinear transform brought by the use of geodesic distances.

The fourth example is a cylinder (Fig. 2 bottom right) and confirms the usefulness of stretched links. The data set contains 20,000 points generated by the parametric equations

$$x = \cos(2\pi v), \tag{11}$$

$$y = \sin(2\pi v), \tag{12}$$

$$z = 4(0.5 - u), \tag{13}$$

where u and v vary between 0 and 1. The results of Isomap and CDA are presented in Fig. 9 (500 prototypes after vector quantization for both methods). Unfortunately, Isomap works difficultly with this non-Euclidean manifold (see the residual variances in Table 1), mainly because it is circular. Visually, Isomap crushes the cylinder like an empty can. Again, CDA manages the problem by tearing the cylinder: once it is cut, the dimensionality reduction is made much easier.

5.2. Image processing

This section still deals with artificially generated data, but now the examples are more realistic in the sense that there is an important dimensionality reduction to achieve. In the field of image processing and visual perception, where the two examples below are coming from, it is not uncommon to retrieve only a few latent variables hidden

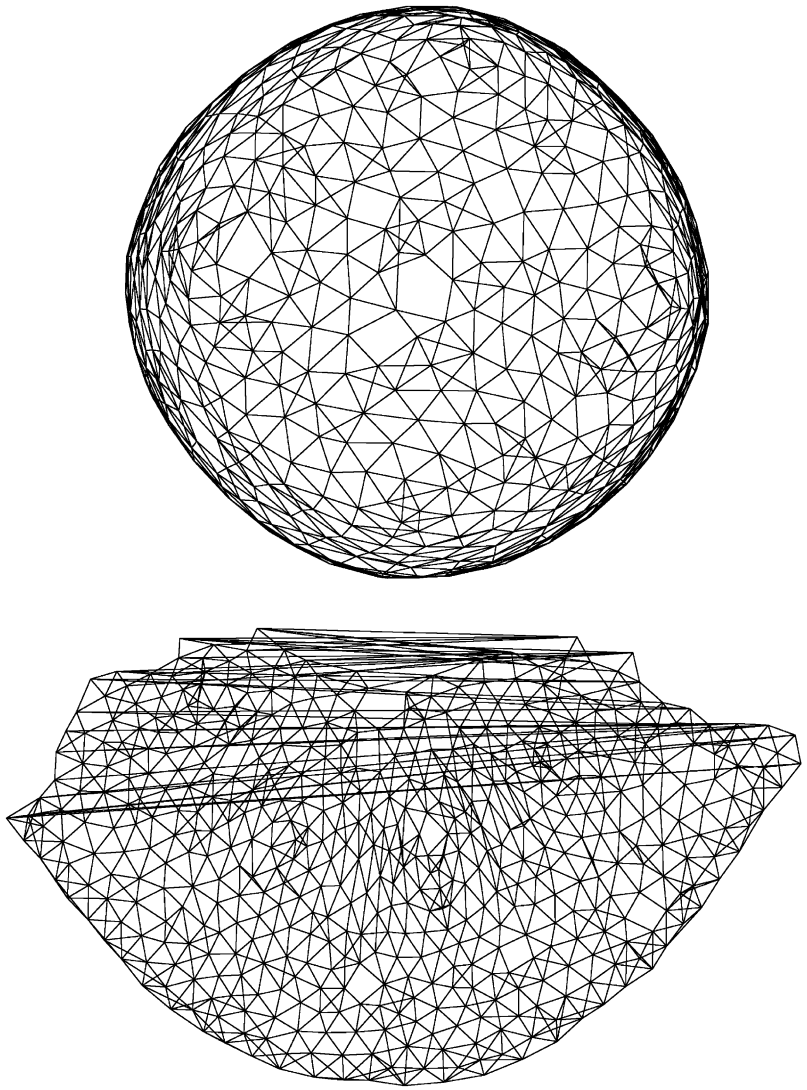

Fig. 8. Two-dimensional projections of 20,000 samples drawn from the open box manifold by Isomap (top) and CDA (bottom); the 500 shown points are the prototypes determined by CDA, which are directly fed into Isomap instead to select 500 points randomly.

among thousands of pixels. Two problems are presented. The first example consists in finding the orientation of a human face in a picture and the second one in reading the time on a traditional (not digital) clock. In both cases, the input data are a set of 4096-dimensional vectors, representing the brightness of 64 pixel $\times$ 64 pixel black-and-white images.

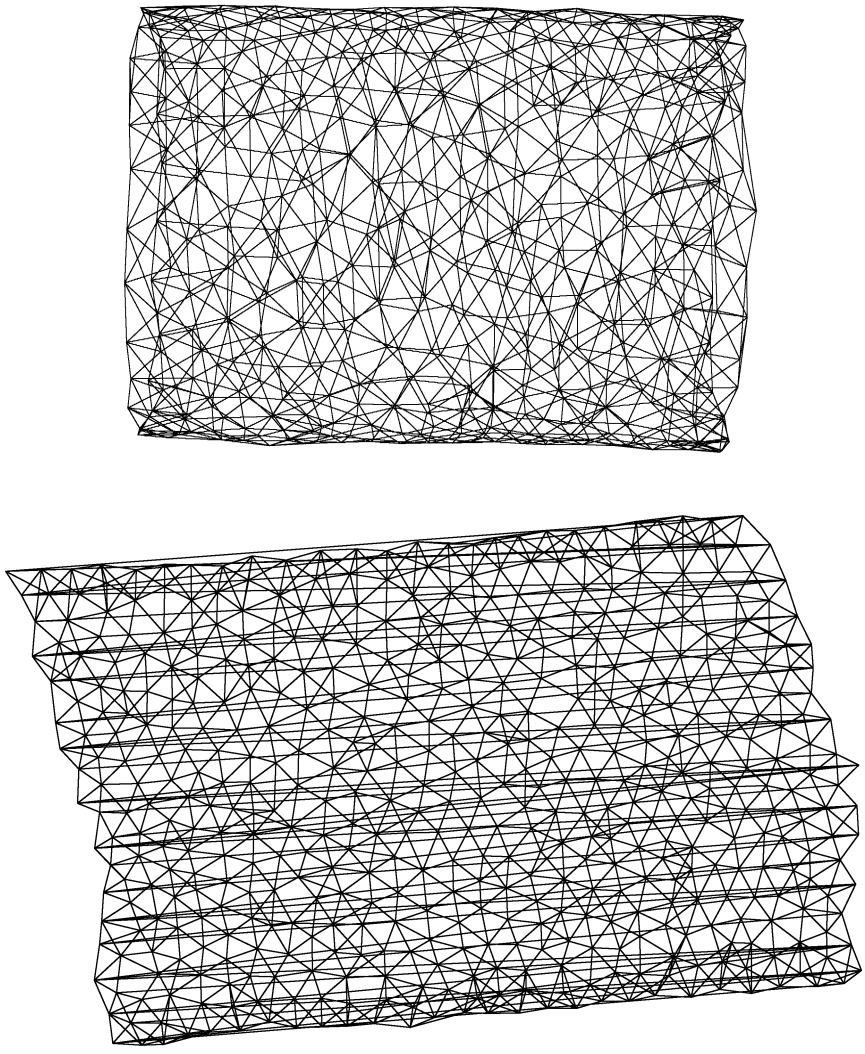

Fig. 9. Two-dimensional projections of 20,000 samples drawn from the cylinder manifold by Isomap (top) and CDA (bottom); the 500 shown points are the prototypes determined by CDA, which are directly fed into Isomap instead to select 500 points randomly.

The face orientation problem is taken from [20] and includes 698 images. Only three variables suffice to distinguish them: the horizontal angle (left–right pose), the vertical angle (up–down pose) and the illumination angle. As advised in [20], the data has been preprocessed by Principal Component Analysis (PCA) in order to reduce the 4096 pixels to only 240 principal components. Afterwards, all images are given to Isomap, without any subset selection. The geodesic distances are computed

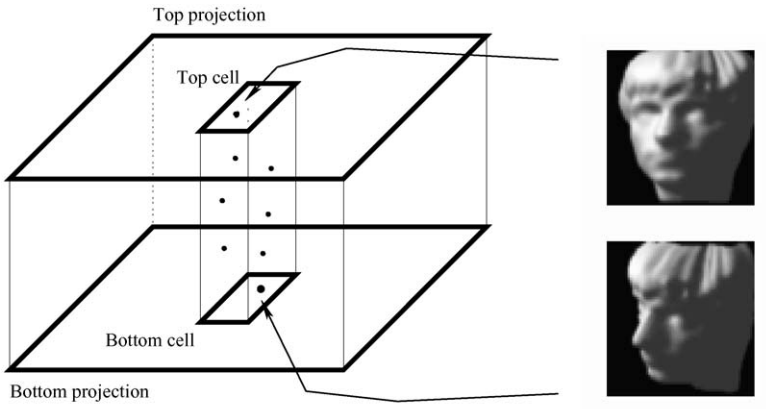

Fig. 10. Projection of the face data: the three-dimensional projection of the face data is divided in cells; in each cell, the highest and the lowest points (bold circle) are displayed in Fig. 11 and 12 with the corresponding two images, respectively on the top and bottom of the figures.

after connecting the input data with $k = 6$. The eigendecomposition of the distance matrix yields the residual variances of Table 1. Since they are negligible and hardly varying starting from the third value, the three eigenvectors associated with the three largest eigenvalues are kept as advised in [20]. This gives a three-dimensional representation of the images; Fig. 10 explains how it is displayed in only two dimensions.

As well for Isomap (Fig. 11) as for CDA (Fig. 12), the 3D representation is divided in 8×8 cells along two dimensions. On the top of each figure, each cell is filled with the image corresponding to the lowest point of the three-dimensional representation, according to the third dimension. On the bottom of each figure, each cell is filled with the image corresponding the highest point lying in the cell.

Obviously, Isomap retrieves the latent variables that have generated the images. Going from left cells to right cells changes the left–right pose while going from top to bottom changes the up–down pose. The illumination direction is full left for the projection of the lowest points and full right for the projection of the highest point in each cell.

Without any care for the parameterization, CDA can deliver similar results, as illustrated in Fig. 12. CDA ran without vector quantization and with $k = 6$ in order to compare fairly with Isomap.

Other experiments have shown that the face orientation problem is a typical example where the geodesic or curvilinear distance plays an essential role. Neither classical metric MDS nor CCA achieve a good unfolding of the manifold with only the Euclidean distance (although a careful parameterization of CCA would maybe lead to acceptable results). Moreover, the manifold seems to be a Euclidean one, as almost no difference appears between Isomap and CDA.

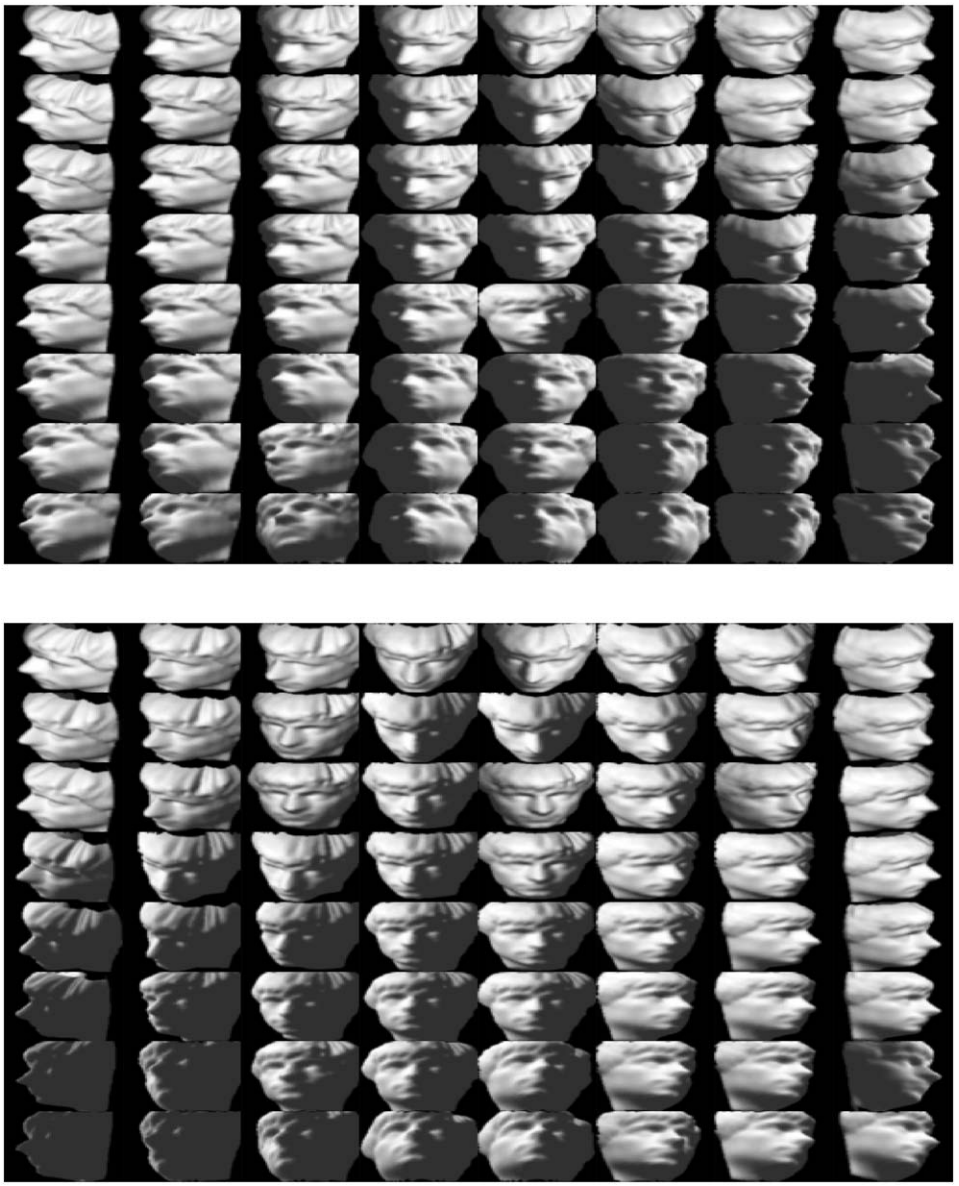

Fig. 11. Projection of the face data set by Isomap (698 faces, no subset selection); the three-dimensional projections is displayed as two 2D projection according to the scheme of Fig. 10.

Such a conclusion, however, does not hold for the second image processing example, for which Isomap and CDA behave differently. The clock reading problem comprises 720 images (one image every minute during 12 h). Therefore, as well Isomap as CDA should theoretically be able to project the 4096-dimensional vectors to only

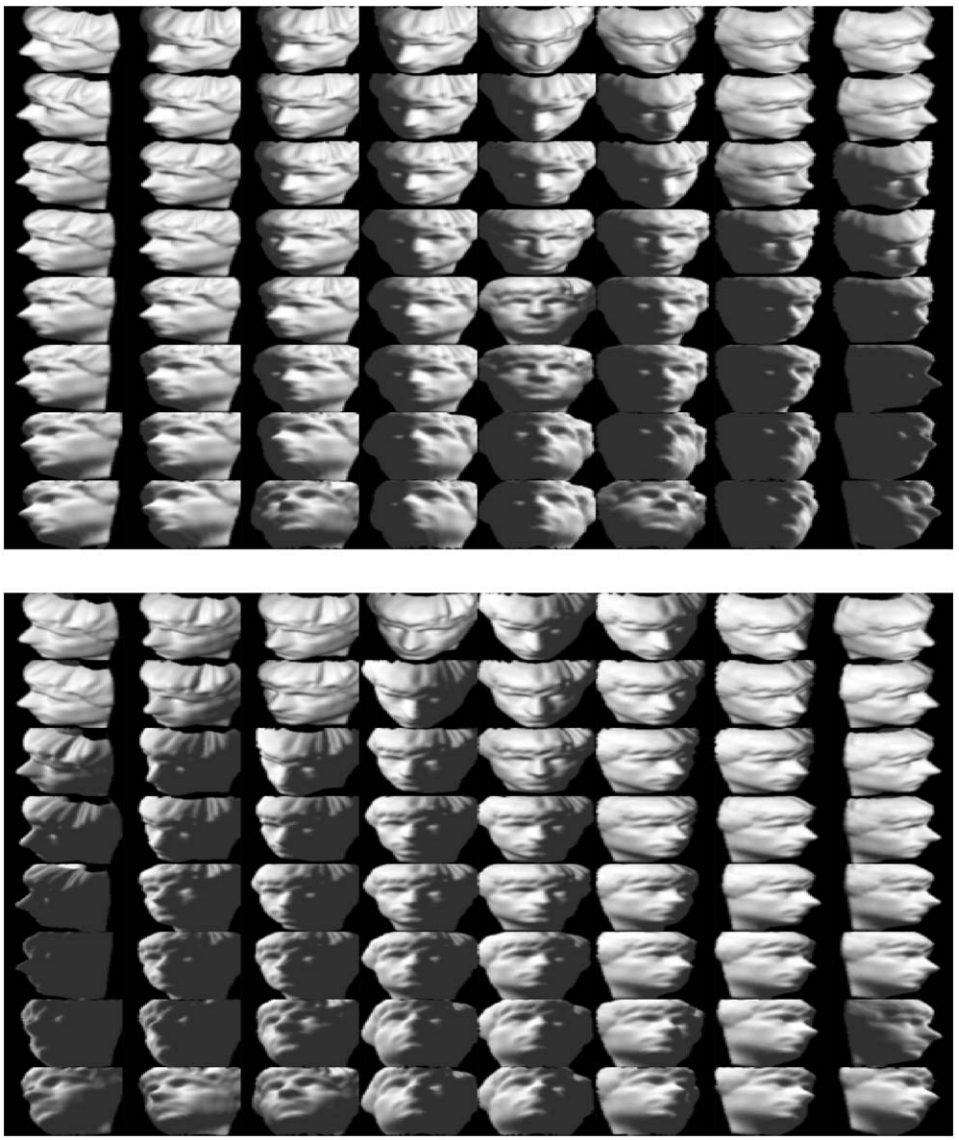

Fig. 12. Projection of the face data set by CDA (698 faces, no vector quantization); the three-dimensional projections is displayed as two 2D projection according to the scheme of Fig. 10.

one variable: the time displayed by the clock![3] But the problem presents a subtlety by comparison with the face orientation. Indeed, assuming the distance between two

[3] Of course, the result will not exactly equate the real time: a shift and a magnification factor have to be taken into account.

consecutive images is smaller than images separated by a longer duration, the manifold is circular, since just after 11h59, the clock goes back to 00h00.

Contrarily to the face orientation problem, Isomap and CDA are fed with the raw data, without preprocessing by PCA. When running Isomap without subset selection and with parameter $k=2$, the residual variances are those mentioned in Table 1. Isomap does not succeed to detect that the underlying manifold is only one-dimensional, since the plateau begins only after the first residual variance. Indeed, when keeping two eigenvectors as indicated in Table 1 and advised by [20], the 2D representation appears to be a circle. Consequently, the 1D representation is inevitably the projection of this circle on a line. Actually, the 1D solution superposes the two halves of the circle, meaning concretely that Isomap cannot distinguish, for example, 06h00 and 12h00. This can be seen in Fig. 13, where the time line, i.e. the 1D projection, has been equally divided by 144 (read from left to right and from top to bottom). Each division is represented by the average of the original images corresponding to the points in the division. The projection of the circle has another shortcoming in the sense that the distribution in the time line is not uniform: the first and last cells include approximately 40 points, i.e. last 40 min, while the cells in the middle of the figure last only 3 min. Visually, this phenomenon can be seen in the figure by observing the minute hand on the clocks: it is well drawn in the middle of the figure, begins to be broader and lighter when moving upwards or downwards, and totally disappears in the first and last cells.

With the help of its gradient descent, CDA computes a totally different solution by cutting and unrolling the underlying circular manifold. The same visual representation as for Isomap is given in Fig. 14. There is no superposition, time goes by as one reads from left to right, row by row, and each division of the time line contains between 4 and 6 points. The minute hand confirms that fact, since its average on a cell is approximately five minutes broad.

The clock reading problem may seem odd or trivial. It is quite easy to write a bunch of code that looks at the right pixels and gives the current time. But unfortunately, the programmer should deeply change his code to adapt it to a new clock. With the approach developed here, the modification is done automatically! If Big Ben is replaced by a Swiss cuckoo clock, one just needs to rerun the mapping algorithm and to recalibrate the obtained time line with the real time (offset, magnification).

6. Discussion

This section briefly gathers some arguments about the way Isomap and CDA solve the problem of nonlinear projection.

6.1. Algebraical versus neural, global versus local

Isomap definitely looks more like an algebraical procedure, resulting from a global approach of the model, while CDA appears rather as a neural network with a local approach of the problem.

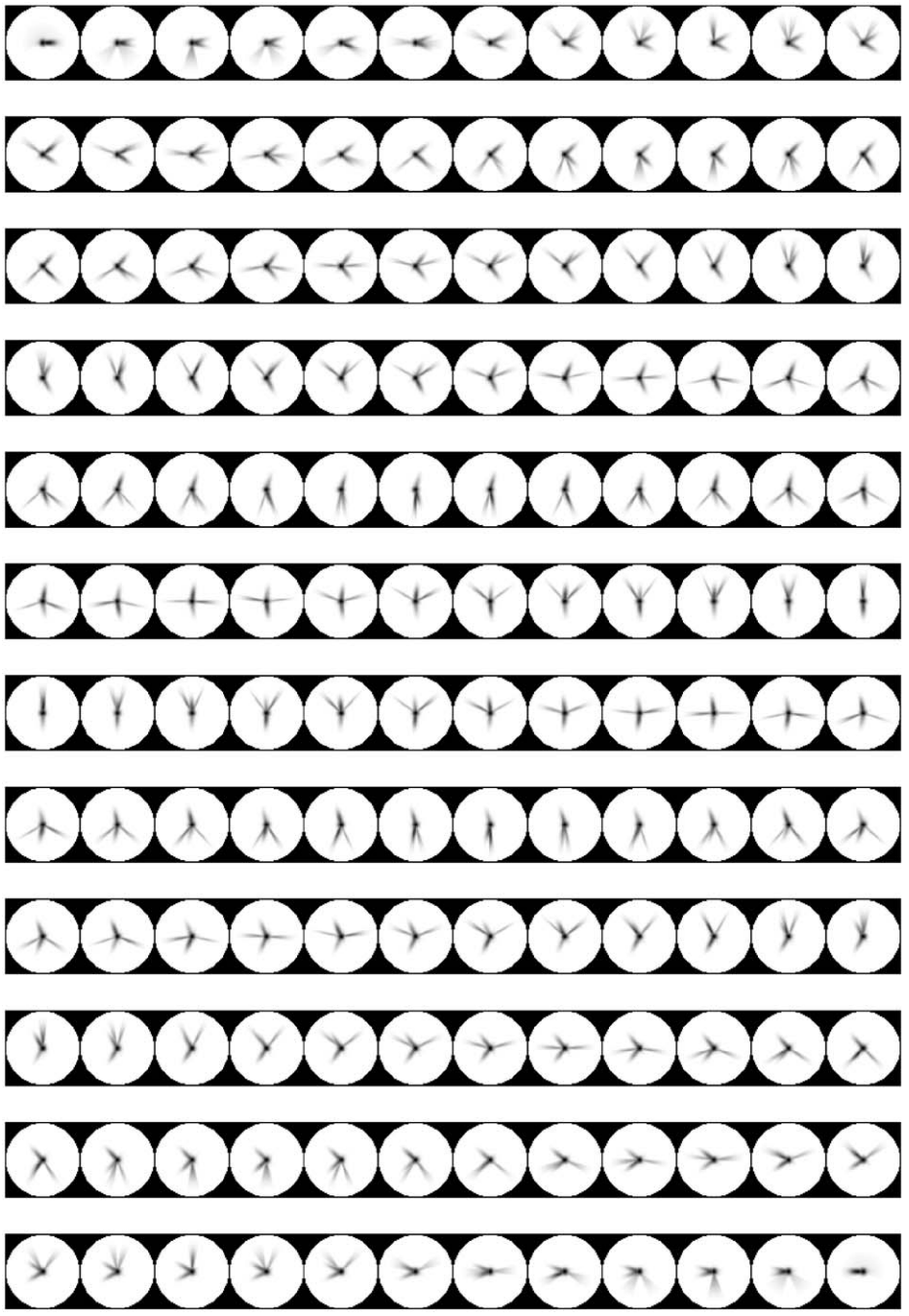

Fig. 13. Projection of the clock data set (720 images) by Isomap (no subset selection) from 4096 to 1 dimension; the projection line is divided in 12 rows to be read from left to right and from top to bottom; each row is then divided into 12 cells filled with the average of the images corresponding to the points lying in the cells.

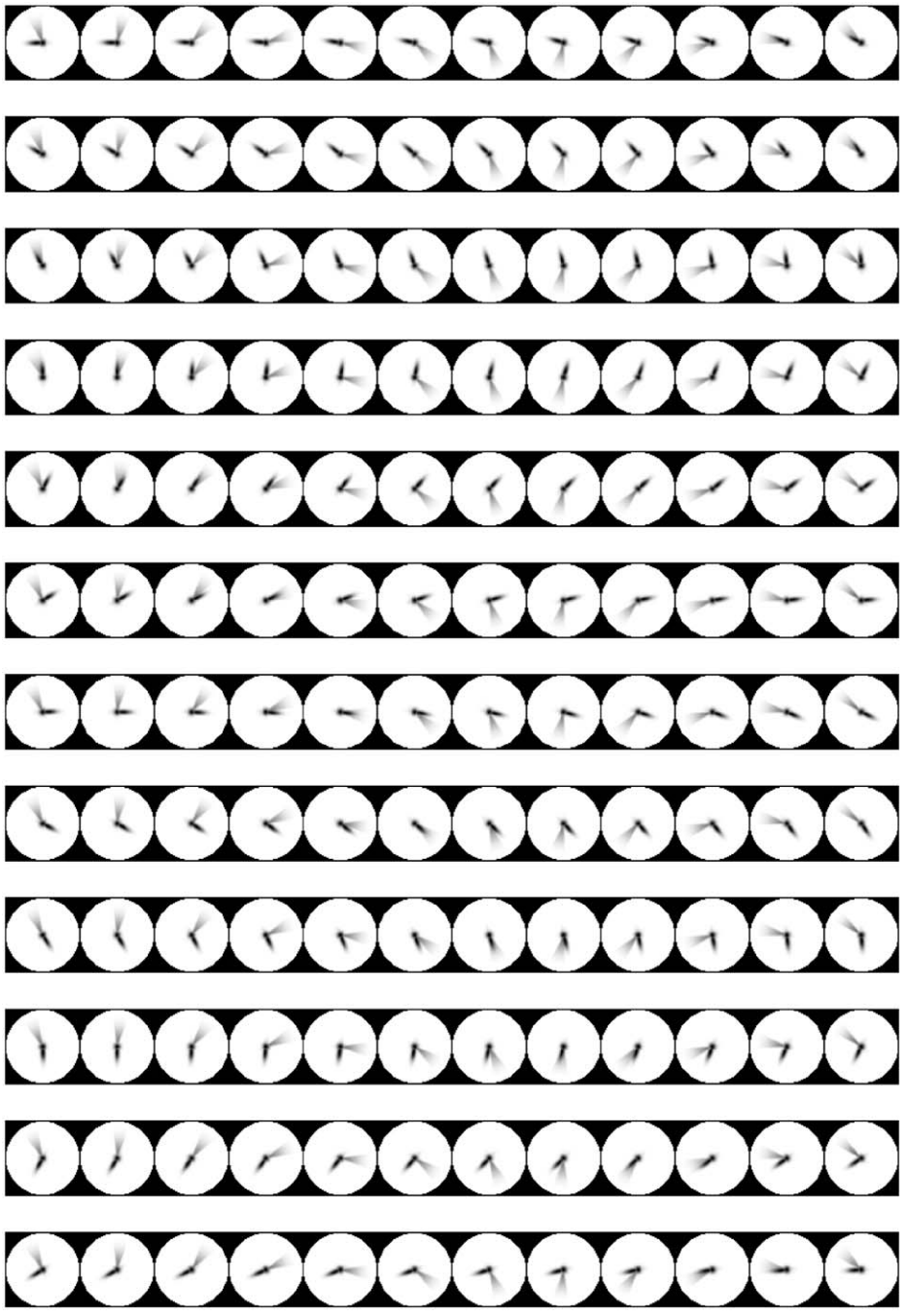

Fig. 14. Projection of the clock data set (720 images) by CDA (no vector quantization) from 4096 to 1 dimension; the projection line is divided in 12 rows to be read from left to right and from top to bottom; each row is then divided into 12 cells filled with the average of the images corresponding to the points lying in the cells.

At first sight, the algebraical approach of Isomap may seem better because it lies on strong theoretical foundations. Nevertheless, algebraical methods, when implemented, often loose some of their beauty due to their translation on real computers, in which floating point numbers are represented with a limited precision. In the case of Isomap, the computation of the eigenvectors of large matrices is not an easy task and it might be critical to rely on a general-purpose algorithm to achieve it. Though, this affirmation need to be moderated, knowing that much work has been decicated to MDS, which faces the same issue. On the other hand, CDA considers the problem from the opposite point of view and solves it directly, without translation or transformation, with an ad hoc procedure, especially tailored for it.

The main advantage of a global approach, at least in the case of Isomap, is that a solution always exists for any problem in the framework of the considered model. This solution minimizes an explicitly formulated criterion. Practically, however, this advantages has its counterparts. The price to pay is often an unrealistic or too constraining model. Similarly, the criterion does not always translate exactly what the user expects. In brief, the solution always exists, is a global minimum, but, when the problem does not fit the model, its interpretation could be hazardous (see Figs. 7, 8 or 9).

On the contrary, a local approach like CDA does not offer any theoretical guarantee. Even when the problem *perfectly* fits the model, an unexperienced user might badly parameterize the algorithm. On the other hand, a well-parameterized local approach is more tolerant and generally delivers a good solution, even if the problem does not fit *exactly* the underlying model. In the case of CDA, the local approach to nonlinear projection consists in focusing the work on small distances and in allowing some distortion for larger ones. In the same way, CDA can accept to shrink some regions of the manifold while stretching (or even tearing) other ones. Obviously, the obtained solution is not guaranteed to be a global optimum, since no explicit criterion exists at the global level. But does the brain optimize some global formula to get good ideas? Probably not.

6.2. Robustness and tolerance

Although Isomap seems to work perfectly on the Swiss roll, this simple manifold, can highlight a major drawback of the method. The case of the parasatic link, detailed above (see Figs. 5 and 6), already underlines the better robustness of CDA. But unfortunately, Isomap does not yield a satisfying solution, even if the connections are well done. In order to notice visually what exactly goes wrong, a thin Swiss roll is needed (it is much longer than wide, see Fig. 15). Using vector quantization, a subset of the points is selected in order to represent correctly the manifold and to avoid shortcut links. Isomap then computes the shortest paths and the low-dimensional coordinates. The residual variances, given in the last row of Table 1, do not decrease clearly, meaning that one dimension already suffices to project the data set. Obviously, the second dimension is small but not negligible and Fig. 16 (top) shows a two-dimensional projection, in agreement with the true intrinsic dimensionality.

Oddly, the projected manifold looks like a bone, wider at both ends than in the middle. What reason could explain such a strange phenomenon? Actually, as mentioned

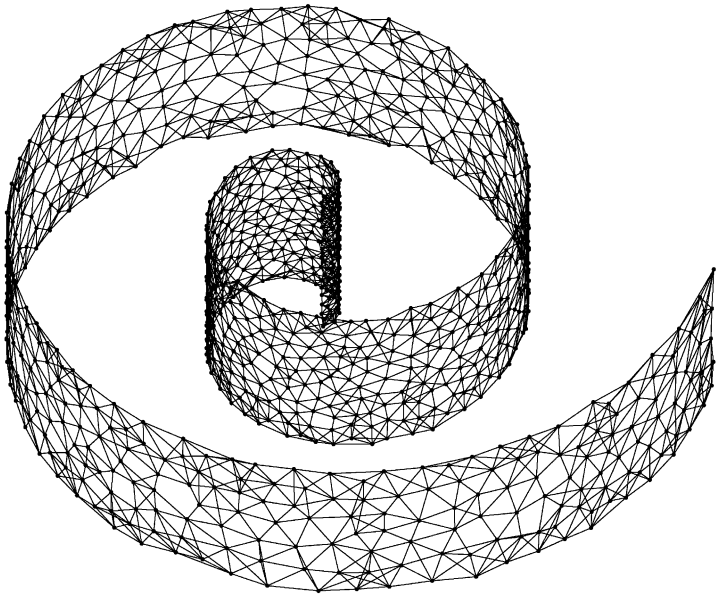

Fig. 15. Thin Swiss roll: 1000 prototypes after vector quantization and linking step ($k = 6$); the length is 4π larger than the width.

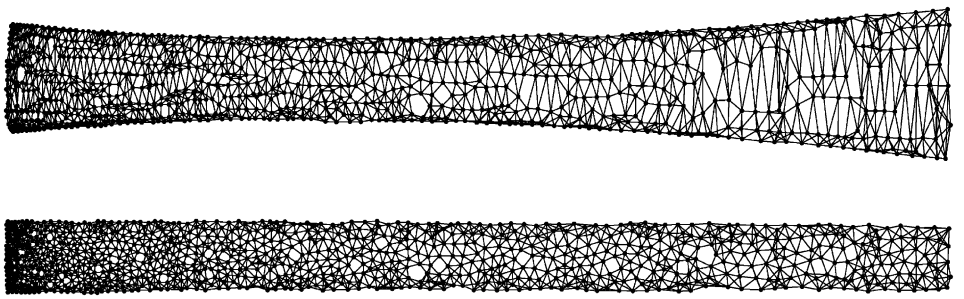

Fig. 16. Thin Swiss roll: two-dimensional projection of the 1000 prototypes of Fig. 15 with Isomap (top) and CDA (bottom).

in Section 2, Isomap only *approximates* the geodesic distances. While the distance is well computed between two linked points, it is unnecessarily overestimated when the shortest path goes through some intermediate points. In this case, indeed, the shortest path usually zigzags [4] on the left and the right, causing the distortion. In the perspec-

[4] At first glance, the overestimation caused by zigzagging could be compensated by the fact that the links do not fit exactly the curvature of the manifold: they are indeed somewhat shorter since they follow a straight line; unfortunately, this concurrent phenomenon *equally* affects *all links.*

Table 2
Mean (and maximum) ratio between the Euclidean distance and the graph distance for n randomly drawn points, connected with the k-rule ($k = 6$) and the ε-rule ($\varepsilon = 0.25$)

n	k-rule	ε-rule
100	1.074169 (1.787279)	1.258154 (6.740509)
200	1.108013 (2.942647)	1.104395 (4.307957)
300	1.124344 (3.680271)	1.037936 (2.771733)
400	1.112438 (3.945047)	1.023038 (2.488945)
500	1.111599 (3.298277)	1.014248 (1.646258)
600	1.115607 (4.165133)	1.010244 (1.403222)
700	1.125792 (4.758089)	1.008191 (1.403222)
800	1.119858 (4.484999)	1.006355 (1.247347)

tive of finding a global solution, this phenomenon is very annoying since this 'noise' pollutes the matrix of all pairwise distances. To avoid the bone-end effect, Isomap should give less importance to large (overestimated) distances and focus on smaller (well estimated) ones, exactly like the flexible neighborhood radius of CDA does. A naive and simplistic solution would be to neglect long distances, but beyond which threshold? And which value would be given in the distance matrix for the neglected entries? How to compute the eigenvectors of such a sparse matrix? Experiments made with CCA/CDA have shown that although long distances are somewhat distorted, they give essential information that is absolutely necessary for the unfolding of objects like the Swiss roll.

Fig. 16 (bottom) shows the result of CDA. Actually, because of the nature of the weighting factor F in $E_{\mathrm{CCA/CDA}}$, CCA/CDA reacts better and more robustly against stretched distances than against shrinked ones.

Another solution to attenuate the bone-end effect in Isomap would be to increase the number of points in the data set, as advised by the theorems proved in [2]. Even when this is possible (the amount of available data is sufficient and the computing time remains reasonable), it experimentally appears that the hypotheses of the theorems are difficultly met, at least for the k-rule. A simple experiment consists in randomly drawing n points uniformly distributed in a square. After connecting them with the k-rule ($k = 6$) or the ε-rule ($\varepsilon = 0.25$), the graph distances are measured. In the case of a plane unit square, the true manifold distances are simply the Euclidean distance. Then, the distortion between the true manifold distances and the graph distances can be computed for several value of n, as it is shown in Table 2. The addition of points does not really improve the distortion for the k-rule. Actually, to get a better behavior, k should grow simultaneously with the number of points. In this case, many links originate from each point in many directions, attenuating the zigzag effect. This is exactly what occurs with the ε-rule when the number of points increases. Unfortunately, a large k or ε implies to build long and numerous links, often incompatible with the curvature of a manifold. Indeed, when the total number of links tends to $n(n-1)/2$, the average number of links to follow along a shortest path tends to one and the geodesic distance reduces to the Euclidean distance. Thus, practically, the projection

of the longer-than-wide Swiss roll requires to connect a huge number of points with the ε rule, ε remaining small to avoid shortcut links. Of course, this situation obliges to solve an enormous eigenproblem with a prohibitive time cost.

Although CDA does not bring a global solution, the result should appear visually more desirable to the user since the manifold is not distorted. In order to get desirable mathematical properties, Isomap pays the price of some compromises and/or restrictions in the definition of the problem.

6.3. Parameterization

Several parameters need to be tuned as well for Isomap as for CDA.

Since both methods share the same procedures to connect the points, some parameters are also common: the value of k or ε according to the choice of either the k-rule or ε-rule. For the mapping stage, only CDA has parameters: α, the learning rate, and λ, the neighborhood radius.

For both methods, the value of k or ε have to be carefully chosen. Too large, they counterbalance the advantages of the curvilinear distance (shortcut links and overall underestimation of the true manifold distance). Too small, they cause the graph distances to overestimate the manifold distance (zigzag effect and unconnected points).

The learning rate of CDA does not need any special care and can be set automatically: a smooth decrease between 1.0 and 0.0 works perfectly. The neighborhood radius λ requires to pay a little more attention for difficult problems, i.e. non-Euclidean manifolds. Intuitively, λ may be seen as a metaparameter acting on the underlying model of CDA. When λ remains large, the model of CDA resembles to the one of Isomap, finding the best global solution with a high probability when the problem perfectly fits the model. On the other hand, when λ is small, the model becomes more tolerant. As mentioned in the preceding subsection, the default parameterization of λ exploits this tolerance to remedy to a not so good choice for k or ε.

7. Conclusion

Isomap and CDA are two nonlinear projection algorithms useful to explore data sets. Both methods work by preserving in the projection the pairwise distances measured in the original manifold, like classical metric MDS does. Although they have been developed independently (and almost simultaneously), they share common innovative ideas. The most important one is probably the alternative metrics they use: the geodesic or curvilinear distance. The new metrics is intended to overcome the difficulties encountered when working with traditional algorithms like metric multidimensional scaling and Sammon's nonlinear mapping. Actually, the geodesic distance facilitates the preservation of the pairwise distances by computing them along the manifold, giving a measure that is almost independent of the manifold curvature.

In this framework, Isomap is just a translation of the classical metric MDS from the language of Euclidean distances into the one of geodesic distances. CDA goes a step

further, by using neural methods (vector quantization, stochastic gradient descent, etc.) like Sammon's nonlinear mapping and CCA.

Because of its mathematical elegance, Isomap has both the advantages of speed and theoretical solidity. Somewhat slower, CDA relies on more complicated techniques and needs to be well parameterized. However, this complexity makes CDA more robust than Isomap on two points. The first one regards the performance of the projection when the manifold does not fit exactly the model hidden behind the geodesic distance (the manifold is not Euclidean). The second point is related to the way geodesic distances are computed: practically, they can only be approximated.

As a global conclusion, the user should remind that an ideal algorithm does not exist for the problem of nonlinear projection. For example, the use CDA or Isomap makes no sense when the data are purely linear: PCA or classical metric MDS will work faster and better. But once the manifold becomes curved, the introduction of the geodesic distance and the choice of Isomap or CDA are fully justified, at least if the manifold remains Euclidean. When even this last condition does not hold, the higher flexibility of CDA is welcome.

References

[1] R. Bellman, Adaptative Control Processes: A Guided Tour, Princeton University Press, Princeton, 1961.
[2] M. Bernstein, V. de Silva, J.C. Langford, J.B. Tenenbaum, Graph approximations to geodesics on embedded manifolds, Technical Report, Stanford University, Stanford, December 2000.
[3] P. Demartines, J. Hérault, Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets, IEEE Trans. Neural Networks 8 (1) (1997) 148–154.
[4] E.W. Dijkstra, A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.
[5] J.H. Friedman, Exploratory projection pursuit, J. Amer. Statist. Assoc. 82 (397) (1987) 249–266.
[6] K. Fukunaga, Intrinsic dimensionality extraction, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Classification, Pattern Recognition and Reduction of Dimensionality, Volume 2: Handbook of Statistics, North-Holland, Amsterdam, 1982, pp. 347–360.
[7] P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors, Physica D 9 (1983) 189–208.
[8] J. Hérault, C. Jaussions-Picaud, A. Guérin-Dugué, Curvilinear component analysis for high dimensional data representation: I. Theoretical aspects and practical use in the presence of noise, in: J. Mira, J.V. Sánchez (Eds.), Proceedings of IWANN'99, Vol. II, Springer, Alicante, Spain, 1999, pp. 635–644.
[9] H. Hotelling, Analysis of a complex of statistical variables into principal components, J. Educ. Psychol. 24 (1933) 417–441, 498–520.
[10] P.J. Huber, Projection Pursuit, Ann. Statist. 13 (2) (1985) 435–475.
[11] A. Hyvarinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley-Interscience, New York, 2001.
[12] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.
[13] N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis, Neural Comput. 9 (7) (1994) 1493–1516.
[14] T. Kohonen, Self-Organizing Maps, 2nd Edition, Springer, Heidelberg, 1995.
[15] J.A. Lee, A. Lendasse, N. Donckers, M. Verleysen, A robust nonlinear projection method, in: M. Verleysen (Ed.), Proceedings of ESANN'2000, Eighth European Symposium on Artificial Neural Networks, D-Facto Publications, Bruges, Belgium, 2000, pp. 13–20.
[16] J.A. Lee, A. Lendasse, M. Verleysen, Curvilinear distances analysis versus isomap, in: M. Verleysen (Ed.), Proceedings of ESANN'2000, Eighth European Symposium on Artificial Neural Networks, D-Facto Publications, Bruges, Belgium, 2002, pp. 13–20.

[17] H. Ritter, T. Martinetz, K. Schulten, Neural Computation and Self-Organizing Maps, Addison-Wesley, Reading, MA, 1992.
[18] J.W. Sammon, A nonlinear mapping algorithm for data structure analysis, IEEE Trans. Comput. CC-18 (5) (1969) 401–409.
[19] D.W. Scott, J.R. Thompson, Probability density estimation in higher dimensions, in: J.R. Gentle (Ed.), Proceedings of the Fifteenth Symposium on the Interface, North-Holland, Elsevier Science Publishers, Amsterdam, New York, Oxford, 1983, pp. 173–179.
[20] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[21] W.S. Torgerson, Multidimensional scaling, I: theory and method, Psychometrika 17 (1952) 401–419.