Nonlinear Data Representation for Visual Learning

Bernard Chalmond, Stéphane Girard

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Nonlinear Data Representation for Visual Learning

Bernard Chalmond, Stéphane Girard

No 3550

Novembre 1998

THÈME 4

ISSN 0249-6399

Nonlinear Data Representation for Visual Learning

Bernard Chalmond, Stéphane Girard

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet is2

Rapport de recherche n° 3550 — Novembre 1998 — 25 pages

Abstract: We are given a set of points in a high dimensional space. For instance, this set can represent many visual appearances of an object, a face or a hand. We address the problem of approximating this set by a manifold in order to have a compact representation of the object appearance. When the scattering of this set is approximately an ellipsoid, then the problem has a well-known solution given by Principal Components Analysis (PCA). Yet, in some situations like object deplacement learning or face learning this linear technique can be ill-adapted and nonlinear approximation must be introduced. The method we propose can be seen as a Non Linear PCA (NLPCA), the main difficulty being that the data points are not ordered. We propose an index to find projection axes encouraging the choice of axes which preserve as well as possible the structure of the closest point neighborhood. These axes determine an order for visiting all the points when smoothing. Finally, a new criterion, called "generalization error" is introduced to determine the smoothing rate, that is the spline number of knots in this case. Experimental results conclude this paper: the method is tested on artificial data and on two data sets coming from databases used in visual learning.

Key-words: Data analysis, Example-based analysis and synthesis, Visual learning, Face representation, Principal components analysis, Nonlinear PCA models, Dimensionality reduction, Multidimensionnal scaling, Projection pursuit, Eingenfeatures.

(Résumé : tsvp)

We thank J.M. Dinten for helpful discussions.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International: +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International: +33 4 76 61 52 52

Représentation Non-linéaire de Données pour l'Apprentissage en Vision

Résumé : Nous avons pour données un ensemble de points dans un espace de grande dimension. Par exemple, cet ensemble peut représenter de nombreuses apparences d'un objet, d'un visage ou d'une main. Nous cherchons à approcher cet ensemble par une variété de façon à obtenir une représentation compacte de l'apparence de l'objet. Lorsque cet ensemble de point est approximativement distribué selon un ellipsoide, le problème a une solution connue donnée par l'Analyse en Composantes Principales (ACP). Cependant, dans certaines situations, comme par exemple l'apprentissage du mouvement d'un objet ou l'apprentissage de visages, cette technique linéaire peut être mal adaptée, et une approximation non-linéaire doit alors être utilisée. La méthode que nous proposons peut être vue comme une ACP non-linéaire, la difficulté essentielle provenant du fait que les données ne sont pas ordonnées. Afin de choisir des axes de projection, nous proposons un index favorisant le choix d'axes conservant aussi bien que possible la structure de voisinage du plus proche voisin. Ces axes déterminent un ordre de parcours des points lors du lissage. Enfin, un nouveau critère, appelé "erreur de généralisation", est introduit afin de déterminer le taux de lissage, c'est à dire le nombre de noeuds des splines dans ce cas. Des résultats expérimentaux concluent ce rapport, la méthode est testée sur données simulées ainsi que sur deux jeux de données issus de bases d'apprentissage pour la vision.

Mots-clé : Analyse de données, Analyse et synthèse basées sur des exemples, Apprentissage en vision, Représentation de visages, Analyse en Composantes Principales, ACP non-linéaire, Réduction de dimension, Poursuite de Projection, Caractéristiques propres.

1 Introduction

In many situations in pattern recognition, machine intelligence, or artificial vision, it is necessary to approximate multidimensional data by a parametric model in order to be able to handle more easily the information contained in these data. In this paper, we keep in mind that the data set $\mathcal{X} = \{x_j,\ 1 \leq j \leq N\}$ belongs to a high dimensional space : $x_j = (x_j^{(i)})_{1 \leq i \leq n} \in I\!R^n$. The problem is then to compress this data set to a low-dimensional manifold. A technique now commonly used for dimensionality reduction in computer vision is Principal Components Analysis (PCA) which yields a linear representation. Since this technique summarizes the data by the mean and the standard deviations (the covariance matrix), the linear representation is accurate only if the data distribution is gaussian, that is, if the cloud $\mathcal{X}$ is a n-D ellipsoid (Fig. 2(a)). Although this linear model is effective in a very broad range of applications, there are however situations where the PCA breaks down. It happens when mean and standard deviations do not accuratly reflect the data distribution, or in other words, when the cloud is not an ellipsoid (Fig. 2(c)). This indicates the need for nonlinear representations.

In this paper a novel technique for designing and fitting nonlinear models is proposed. From the geometric point of view, we try to approximate $\mathcal{X}$ by a d-dimensional manifold with $d < n$ [30]. This manifold will be given by an implicit equation $G(\theta, x) = 0$, where θ is a vector of parameters and $G(\theta, .)$ is a continuously differentiable function from $I\!R^n$ to $I\!R^n$. A model is a particular analytical expression of G, the most simple being the linear representation as it is given by Principal Components Analysis. Concisely speaking, our problem is a smoothing problem. But in our case, it is not a classical smoothing problem since the points x_j *are not ordered*. Fitting of manifolds is a much more difficult problem than function fitting. Here, this difficulty is emphasized by a second one. The number of points x_j in the data set is usually moderate with respect to the dimension n. This means that the space $I\!R^n$ is nearly empty, what it is known as the "curse of dimensionality" [22].

1.1 Application domains

We are going to illustrate our purpose by examples in visual learning although our technique is not limited to this domain. The data set $\mathcal{X}$ are N instances of an object appearance. Here, "object" is a generic term which designates for example a rigid or flexible manufactured object, a face, a hand,... [32, 15, 23, 26]. It is convenient to see $\mathcal{X}$ as a set of occurrences of a random vector X in $I\!R^n$. Such an occurrence can be a curve [16], an image [23], or a feature vector as Gabor coefficients for example [29]. An approach for the visual learning problem consists in fitting a compact model of the object's appearance to $\mathcal{X}$. The aim of this appearance-based representation is to capture the flexibility or the changes of the object in a compact manner such that every occurrence of X can be approximated by a point of the manifold defined by the model. Such a representation has wide applications: recognition and pose of 3D objects [32], face recognition [23, 19, 42], face tracking [29], gesture recognition, [34], image retrieval [43]. In visual learning, PCA is widely used for reducing

dimensionality to enable efficient learning. However, the limitation of the PCA use is limited since its linear representation is inappropriate for modelling nonlinear effects such as bending or rotation of shape. This drawback has been highlighted by Murase and Nayar [32]. These authors have used the principal components of many views of a single rigid object to visualise the low dimensional manifold describing changes due to rotation and illumination conditions. The object's pose could then be determined by its position on this manifold.

Let us say a few words on the application which has motivated our research. In [15], the authors address the problem of identifying the radiographic projection of an object (its appearance) from incomplete data extracted from a radiographic image. They assume that the unknow appearance is a particular instance of the projection of a flexible object. Their approach consists first in learning a deformation model and its probability distribution able to represent and to simulate a great variety of appearances. This modelization is achieved using a training set of complete appearances. Then, given the incomplete data, the identification task consists in estimating the unknown appearance using the previous model whose probability distribution plays the role of a prior distribution in a Bayesian framework.

1.2 PCA and its limitations

Let us recall some important points of the PCA method [28], which will be usefull for the forthcoming nonlinear model. The PCA model is linear. When the data are centered, its expression is:

$$\begin{aligned} G(\mathbf{a},x) &\stackrel{\text{def}}{=} x - F(\mathbf{a},x) \\ \text{with} \quad F(\mathbf{a},x) &= \sum_{k=1}^{d} a_k \, (a_k^T x) \ , \end{aligned} \tag{1}$$

where $a_k \in I\!R^n$ are unit orthogonal vectors which span the principal axes. Then, $G(\mathbf{a},x) = 0$ defines a d-dimensional linear subspace in $I\!R^n$. As the x_j are observations of a non-degenerated random variable, it is clear that they cannot verify the deterministic equation $G(\mathbf{a},x_j) = 0$ all together : $G(\mathbf{a},x_j) \neq 0$ and we define :

$$r_j(d) \stackrel{\text{def}}{=} G(\mathbf{a},x_j) \ . \tag{2}$$

For a d-dimensional model, $r_j(d)$ is interpreted as the residual approximation error of x_j by $F(\mathbf{a},x_j)$. The model (1) is said to be auto-associative since x^j is approximated by a function of $\{x_j\}$. The model parameters $\mathbf{a} = (a_1,...,a_d)$ in (1) are estimated by minimizing the empirical mean square $\sum_{j=1}^{N} \|r_j(d)\|^2$. These estimations are related to the x_j orthogonal projections on the desired subspace and consequently the r_j are the Euclidian distances to the subspace.

Let us consider an artificial data set $\mathcal{X}$ composed of translated curves (Fig.1(a)). This set corresponds to a cloud in $\mathbb{R}^n$ located on a 1-dimensional manifold. We can imagine what this manifold looks like by projecting $\mathcal{X}$ on the first principal plane computed by PCA (Fig.1(d)). This kind of figure is known in the statistical litterature as "the horseshoe effect" [28]. We get a similar phenomenon when $\mathcal{X}$ is composed of faces in rotation [29]. On the artificial training set, the estimated linear model leads to non-realistic curve simulations and does not allow to reproduce the variations due to a translation (Fig.1(c)). Here, the simulated curves are obtained by drawing random points on the principal axes. This very simple academic example is only presented to illustrate our purpose and it must not be mistaken with a registration problem [36]. Even for variation as simple as translation, the linear model given by PCA is ill-adapted. More complex exemples are showned in Fig.6 and Fig.8. Recently, in the context of visual learning for object representation, nonlinear problems have arisen several times, but no general solution has been proposed [6, 32].

1.3 Relationship to previous research

The reconstruction of $n-1$ dimensional manifolds in $\mathbb{R}^n$, like curves in $\mathbb{R}^2$ or surfaces in $\mathbb{R}^3$, seems closed to our problem at first glance. In [21], a numerical algorithm is proposed. Its input is an unordered set $\mathcal{X} \subset \mathbb{R}^3$ near an unknown manifold, and its output is a surface that approximates the manifold. Reconstruction problems of this kind occur in engineering applications like representation from range data. This algorithm makes heavy use of the fact that the data set is "continuously dense" in some sense, around the unknown manifold. This situation is not verified in our case and furthermore, we are firstly interested by manifolds whose dimension d is much smaller than n, and secondly by representations based on an analytical model.

The PCA methods for analysing the variations of flexible curves, have become an own research topic these last years, called "Fonctional data analysis" [36]. The basic philosophy of functional data analysis is that one should think of observed data functions as single entities, rather than merely a sequence of individual observations. Nevertheless, the results of this research have not been generalized to flexible surfaces, i.e. to images.

Besides, PCA generalizations have been proposed in order to take into account nonlinear phenomena. PCA-like auto-associative methods have been studied from the neural networks point of view with perceptron networks [24], but in fact these models remain linear. In the case $n = 2$, a truly nonlinear approach is proposed in [20]. It consists in searching a curve called "principal curve" which passes through the middle of the data set. This means that every point on the curve is the average of the observations projecting onto it. This yields a second approach to the $n-1$ dimensional manifofd reconstruction as introduced above. Yet, this technique is mainly dedicated to the dimension $n = 2$ and is non-parametric. Its extension to a greater dimension is a difficult task.

A specific technique closed to PCA is Multimensional Scaling (MDS) [3, 27, 39]. MDS is concerned with the problem of constructing a configuration of N points $\mathcal{Y} = \{y_j,\ 1 \leq j \leq N\}$ in $\mathbb{R}^d$ satisfying $\|y_i - y_j\| \approx \|x_i - x_j\|$. This is achieved by minimizing a distorsion index whose expression is for instance [39]:

$$I_S(\mathcal{Y}) \quad = \quad \frac{\sum_{i<j}(\|y_i - y_j\| - \|x_i - x_j\|)^2}{\sum_{i<j}\|y_i - y_j\|^2} \ . \tag{3}$$

In this spirit, MDS technique has been applied to build self-organized neural networks which give a low-dimensional mapping of the submanifod of a high-dimensional and non linearly related data set. It yields an unfolding of the data submanifold. We shall come back later to that kind of index.

In the second section of this paper, we define a Non Linear Principal Component Analysis (NLPCA) that we limit to one axis ($d = 1$) for sake of clarity. Our aim is to approximate $\mathcal{X}$ by a n-dimensional curve C. In other word, we search to determine a curve C which estimates an ideal curve C^* using the sample $\mathcal{X}$. From this point of view, we have to deal with a curve reconstruction problem. Curve reconstruction has efficient solutions when the x_j are ordered and a parametrization of the curve is given. So, like for MDS technique, our approach consists in searching a parametrization axis which yields an approximate ordering of the points x_j. It follows that our methods shall work well when the curve to reconstruct is, roughly speaking, *the graph of a function over an unknown axis* in $\mathbb{R}^n$. In $\mathbb{R}^2$, this approach is less general than the principal curves one. In the third section, this analysis is extended to several axis ($d \geq 1$). Finally, in the fourth section, experimental results will be described.

2 Nonlinear PCA with one axis

2.1 Introduction

Let us gradually introduce our algorithm in an intuitive fashion using Fig.2. In Fig.2(a), $\mathcal{X}$ is a non-isotrope Gaussian cloud and thus PCA is well adapted. Each point x_j in $\mathcal{X}$ is approximated by its projection on the principal axis defined by the unit vector a_1. This projection is:

$$F(a_1, x_j) \quad = \quad a_1(a_1^T x_j) \ .$$

In Fig.2(b,c,d) the cloud $\mathcal{X}$ is no more Gaussian and thus its approximation by projection on the pincipal axis is inaccurate. In Fig.2(b), we are naturally led to approximate $\mathcal{X}$ by a smooth curve with parameter b_1. In that case, the x_j approximation is:

$$F(\theta_1, x_j) \ = \ S_{b_1}\left(a_1^T x_j\right) \qquad \text{with} \qquad \theta_1 \ = \ (a_1, b_1) \ .$$

S is a function of $u = a_1^T x$. Let us note that $u_j = a_1^T x_j$ is the coordinate of the x_j projection on the axis a_1. On this figure, the order of the u_j in $\mathbb{R}$ approximatively corresponds to the

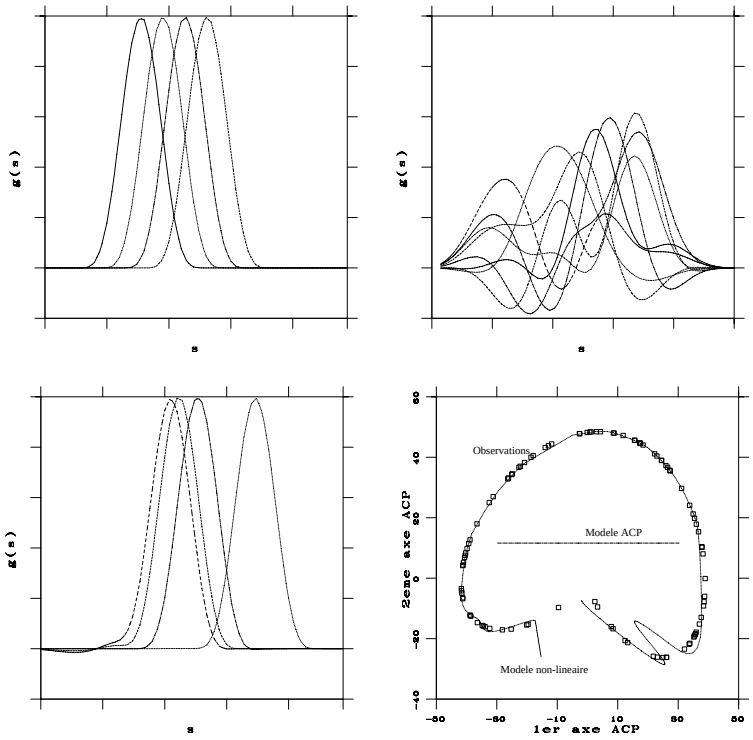

Figure 1: (a) Translated curves. (b) Simulation using PCA. (c) Simulation using NLPCA. (d) Projection on the first PCA plane.

"topological order" of the x_j in $\mathbb{R}^n$: this is a crucial point. On the contrary, in Fig. 2(c) the principal axis a_1 does not allow to determinate the topological order. For instance, two non neighbor and distant points in $\mathbb{R}^n$ may have neighbor projections on the axis a_1. It is a crucial problem. If we still attempt to approximate $\mathcal{X}$ by a smooth curve parametrized by $u = a_1^T x$, the representation would be inadequate for synthesis: outside the values u_j, the function $S_{b_1}(u)$ would not deliver realistic occurrences of X (Fig.2(c)). This difficulty can be overcomed by choosing an axis which preserves as well as possible the topological order after projection (Fig.2(d)). For this, we shall define an index $I(a, \mathcal{X})$ measuring the preservation of the neighborhood structure of $\mathcal{X}$. The selected axis will maximize this index:

$$a_1 = \arg\max_a I(a, \mathcal{X}) \tag{4}$$

Finally, the smooting function parameter is obtained as follows:

$$\begin{aligned} b_1 &= \arg\min_b \frac{1}{N} \sum_{j=1}^{N} \|r_j(1)\|^2 \\ \text{with} \quad r_j(1) &= x_j - S_b\left(a_1^T x_j\right) \ . \end{aligned} \tag{5}$$

Our curve fitting algorithm consists of two stages: axis search (4) and smoothing (5). The next several sections develop in more details the successive steps of the algorithm.

2.2 Definition of I and S

In (4) and (5), we have to define the projection index $I(a, \mathcal{X})$ and the smoothing function S_b.

Projection index

A first idea would be to use the index (3) which attempts to preserve all the distances. In our context, the important point is not the distance preservation but the preservation of the neighborhood structure of $\mathcal{X}$: if two points are neighbors in $\mathbb{R}^n$ then their projections should be neighbor in $\mathbb{R}$ and conversely, as it is illustrated in Fig.2(d) but not in Fig.2(c) for many points.
To simplify this criterion, we decide not to preserve the neighborhood structure entirely, but only the *closest neighbor* structure. Yet, it is usually impossible to preserve the whole neighborhood structure. So, the chosen index is naturally the number of points for which the constraints are verified:

$$I(a, \mathcal{X}) = \sum_{i=1}^{N} \sum_{j \neq i} \Psi[(x_j \text{ closest to } x_i) \implies (a^T x_j \text{ closest to } a^T x_i)] \ , \tag{6}$$

where Ψ is the indicator function. The analytical expression of I is obtained as follows. Let $x_{\phi(i)}$ be the closest neighbor of x_i. We want $a^T x_{\phi(i)}$ to be the closest neighbor of $a^T x_i$ in

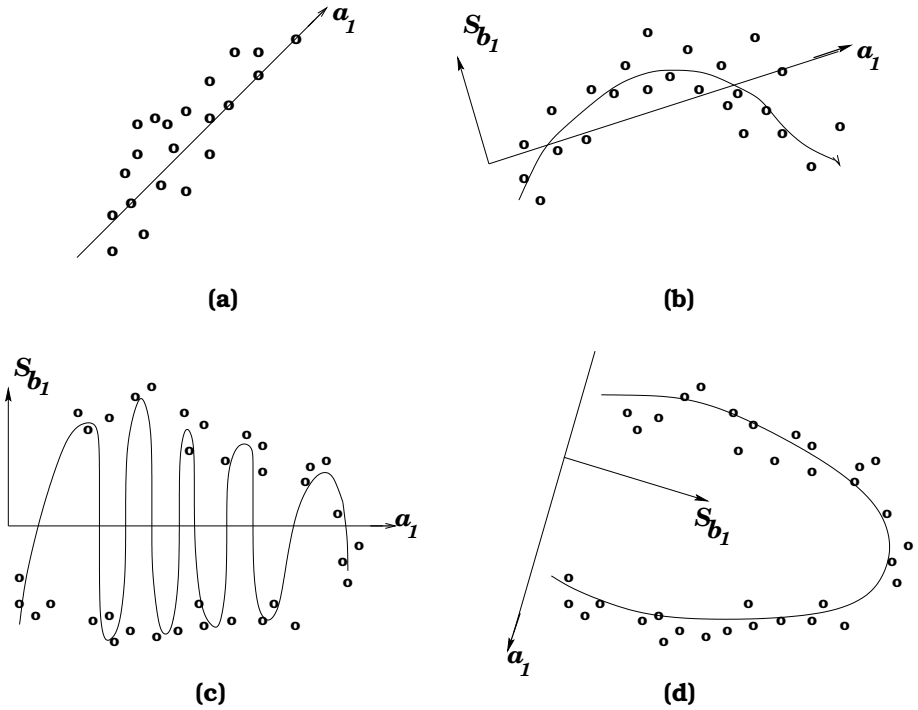

Figure 2: (a) Gaussian cloud. (b-c-d) Non Gaussian cloud.

$I\!R$. Consequently, for every i, a has to verify $N-1$ inequalities:

$$\begin{aligned} I(a,\mathcal{X}) &= \sum_{i=1}^{N}\sum_{j\neq i}\Psi\left[\ \left|a^T x_i - a^T x_{\phi(i)}\right| \leq \left|a^T x_i - a^T x_j\right|\ \right] \\ &= \sum_{i=1}^{N}\sum_{j\neq i}\Psi\left[\ \left|a^T \bar{x}_{i\phi(i)}\right| \ \leq\ \left|a^T \bar{x}_{ij}\right|\ \right]\ , \end{aligned}$$

where $\bar{x}_{ij} = x_i - x_j$. Finally, since a and $-a$ define the same axis, we get:

$$I(a,\mathcal{X}) = \sum_{i=1}^{N}\prod_{j\neq i}\Psi\left[a^T\left(\bar{x}_{i\phi(i)} - \bar{x}_{ij}\right)\geq 0\right]\times\Psi\left[a^T\left(\bar{x}_{i\phi(i)} + \bar{x}_{ij}\right)\leq 0\right]\ . \tag{7}$$

It can be showned this index has the following invariance properties: *(1)* $I(a,\mathcal{X}) = I(a,\mathcal{X}+t)$, $t\in I\!R^n$; *(2)* $I(a,s\mathcal{X}) = I(a,\mathcal{X})$, $s\in I\!R$; *(3)* $I(Da,D\mathcal{X}) = I(a,\mathcal{X})$ with ${}^tDD = I$. The first two invariance properties with respect to translation and scale indicate that this index belongs to the class III defined by Hubert [22], which is well-adapted for Projection Pursuit algorithms. The last property shows that the search for the axis does not depend on the orientation of $\mathcal{X}$ (rotation and symmetry invariance).

Smoothing function

The choice of S_b is much more simpler than the index choice. We express S_b as a multivariate spline function $S_b : I\!R \to I\!R^n$ [10]. This representation is well-known. The i-th coordinate of S^b is noted $S^{(i)}_{b^{(i)}}$ with $b = (b^{(i)})_{1\leq i\leq n}$, We approximate the ith coordinate $\{x_1^{(i)},...,x_N^{(i)}\}$ of the vector set $\mathcal{X}$ with respect to the parametrization $\{u_1,...,u_N\}$, by a cubic regression spline $S^{(i)}_{b^{(i)}}(u)$, $u\in[u_{\min},u_{\max}]$. To do that, for each coordinate i, we need to sort the set $\{u_j\}$. Let $\{u_{\sigma(j)}\}$ be the sorted set. The i-th coordinate smoothing is performed on the couples $\{(u_{\sigma(j)},x^{(i)}_{\sigma(j)}),\ j=1,...,N\}$. Such a spline representation implies to choose discontinuity knots in the $u_{\sigma(j)}$ series. The number of knots tunes the smoothing rate. In general, this choice is quite difficult and will be discussed in Appendix.

2.3 Computation

Index maximization

To perform this optimization, we have to note that $I(a,\mathcal{X})$ is a finite piecewise constant function. Since the equation $a^Tx = 0$ implies that a belongs to the hyperplane orthogonal to x, the $I(a,\mathcal{X})$ expression (7) shows $N(N-1)$ hyperplanes noted H_k:

$$\begin{aligned} H_k &= \left\{a\in I\!R^n \ \ s.t.\ \ a^Tn_k = 0\right\},\quad 1\leq k\leq N(N-1), \\ n_k &\in \left\{\left(\bar{x}_{i\phi(i)} - \bar{x}^{ij}\right)\ ,\left(\ \bar{x}_{i\phi(i)} + \bar{x}^{ij}\right)\ ;\ i=1,...,N\ ;\ j\neq i\right\}\ . \end{aligned}$$

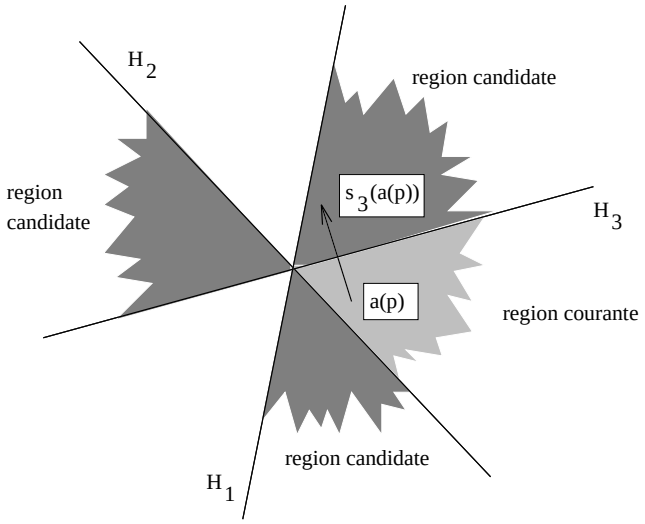

Figure 3: Random walk on the regions.

These $N(N-1)$ hyperplanes determine a $\mathbb{R}^n$ partition in L regions R_ℓ, $(L \leq 2^{N(N-1)})$, in which the index is constant: $I(a, \mathcal{X}) = \sum_{\ell=1}^{L} I_\ell \ \Psi[a \in R_\ell]$.
Although it is possible to compute I for every vector a by applying (7), we do not know how to extract a vector a from every region R_ℓ. To overcome this difficulty, the idea is to construct an iterative algorithm A of type $a(p+1) = A[a(p)]$ which visits the regions from an initial solution $a(0)$ (as a gradient algorithm would do it in the case of a regular function), $a(p)$ and $a(p+1)$ being in distinct regions. The framework of the stochastic algorithms yields a general formalisation to that optimization problem.
The idea to maximize I is now to build a random walk upon the regions in order to try to increase the index value at each step. Each step is defined by a hyperplane H_k obtained by a random choice of the integers k and considered as a border of the new visited region. If $a(p)$ is the solution at the previous step, the new candidate $a(p+1)$ is obtained by the orthogonal symmetry s_k with respect to H_k: $s_k(a(p+1)) = a(p) - 2(a(p)^T n_k)n_k$.

Algorithm A

1. Initialization: $a(0)$ is chosen randomly.
2. $a(p)$ is known, $a(p+1)$ is computed as follows :
   - k is chosen randomly between 1 and $N(N-1)$.
   - Comparison between the regions separated by H_k:
     $a' \leftarrow s_k(a(p))$.
     If $I(a', \mathcal{X}) > I(a(p), \mathcal{X})$
     Then $a(p+1) \leftarrow a'$ Else $a(p+1) \leftarrow a(p)$.
3. Go back to 2, until convergence.

The resulting index sequence is decreasing, the algorithm converges quickly towards a *local* maximum. An iteration cost isequivalent to the index $I(a', X)$ computation cost, that is $\mathcal{O}(nN^2)$ (differences $\bar{x}^{ij}$ are computed as a preprocessing at the intialization step).

Algorithm A'

When a local maximum has been reached, all the regions accessible by the operator s_k (dark grey regions in Fig.3) have an index value lower than the maximum one (clear grey regions in Fig.3).Yet, the index value on the other regions (white regions in Fig. 3) could be greater than the local maximum. As it is impossible to reach directly these regions from the region of local maximum, a solution for visitingthem is to pass by a region of lower value (a dark grey region), the transition being governed by a probability. This is the basic idea of the simulated annealing that we briefly describe in our context [35]. Contrary to the algorithm A which imposes at each iteration p that $\Delta I \doteq [I(a(p+1), \mathcal{X}) - I(a(p), \mathcal{X})] > 0$, the stochastic algorithm (algorithm A') authorizes ΔI to be negative: $\Delta I > T_p \log \xi$ where ξ is a uniform random number on $]0,1[$ and T_p is a sequence decreasing towards 0 : $T_p = T_0 \ \lambda^p$, with $\lambda < 1$. The algoritm A' is the algorithm A in which the condition If...Then is modified by plugging $\Delta I > T_p \log \xi$. As p grows $T_p \log \xi \to 0$, and the probability to accept a region of lower index becomes zero. A rigorous writting of this algorithm can be derived from the Metropolis dynamic [5].

Spline smoothing

At this time, the principal axis is known through its estimation a_1 as obtained above. We have now to deal with a well-known smoothing problem since we use a spline regression model. Let us denote again $u_j = a_1^T x_j$ the coordinate of the projection of x_j. The goal

here is to find b_1 which minimizes the approximation error :

$$\begin{aligned}\epsilon^2(b) &= \frac{1}{N}\sum_{j=1}^{N}\|r_j(1)\|^2 = \frac{1}{N}\sum_{j=1}^{N}\|x_j - S_b(u_j)\|^2, \\ &= \frac{1}{N}\sum_{i=1}^{n}\left[\sum_{j=1}^{N}\left(x_j^{(i)} - S_{b^{(i)}}^{(i)}(u_j)\right)^2\right] = \sum_{i=1}^{n}\epsilon_i^2(b^{(i)}). \end{aligned} \tag{8}$$

It appears that the approximation error ϵ^2 is expanded into n independent approximation errors ϵ_i^2. Let us note B the B-spline matrix corresponding to the chosen knots. Each of the n approximation errors is then written as :

$$\sum_{j=1}^{N}\left(x_{\sigma(j)}^{(i)} - S_{b^{(i)}}^{(i)}(u_{\sigma(j)})\right)^2 = \|x^{(i)} - Bb^{(i)}\|_N^2, \quad 1 \le i \le n\ , \tag{9}$$

where $\|.\|_N$ is the $\mathbb{R}^N$ Euclidian norm. Let $b_1^{(i)}$ be the least-squares minimum, and ν the spline number of knots. Let us emphasize that $b_1^{(i)}$ is computed as soon as ν is fixed, hence is denoted as $b_1^{(i)}(\nu)$. We use a second criterion to determine the best number of knots, as presented in Appendix.

3 Nonlinear PCA with several axes

In the spirit of the Projection Pursuit [22], a second axis of projection can be computed on the residuals $r_j(1)$ which take the place of $\mathcal{X}$ in (4) and (5), leading to new residuals $r_j(2)$. This procedure can be again performed on the new residuals $r_j(2)$, and so on.

NLPCA algorithm

1. Initialization:
   $k \leftarrow 0$ and $r_j(0) \leftarrow x_j \ , \ j = 1, ..., N$
2. Parameter estimation: $\theta_{k+1} = (a_{k+1}, b_{k+1})$

$$a_{k+1} = \arg\max_a I(a, \{r_j(k)\}_{j=1}^N) \tag{10}$$

$$b_{k+1} = \arg\min_b \sum_{j=1}^N \|r_j(k) - S_b(a_{k+1}^T \ r_j(k))\|^2 \tag{11}$$

3. Residual errors update:
   $r_j(k+1) \leftarrow r_j(k) - S_{b_{k+1}}(a_{k+1}^T \ r_j(k)) \ , \ j = 1, ..., N$
4. Go back to 2, if the residual errors are too large, with $k \leftarrow k+1$;
   else $d = k$ and end.

Orthogonality conditions

We achieve the algorithm by adding some constraints that we have not given so far in order to make the presentation clearer. Up to now, we have implicitly supposed that the residual errors $r_j(k)$ are orthogonal to the axis a_k:

$$a_k^T r_j(k) = 0 \ , \quad \forall j \ . \tag{12}$$

It is straightforward to show that (12) follows the natural condition $a_k^T S_{b_k}(u) = u$. Thanks to this condition the algorithm is very simple. After an approximation with respect to the axis a_k, the residual errors $r^j(k)$ are located in its orthogonal subspace. So, the new axis as well as the new smoothing function are built in the $a_k, a_{k-1}, ..., a_1$ orthogonal subspace, that is:

$$\begin{aligned} a_k^T a_\ell &= 0 \ , \qquad \forall \ 1 \leq \ell, \ k \ \leq d \\ a_\ell^T S_{b_k}(u) &= 0 \ , \qquad \forall \ 1 \leq \ell < k \ \leq d \ , \quad u \in I\!R \ . \end{aligned} \tag{13}$$

Let us emphasize that the orthogonality condition (13) gives an answer to the difficult problem of iterating the Projection Pursuit algorithm from a_k to a_{k+1}, [11].

Associated model

Now, the question is: what kind of representation is produced by our algorithm ? For $d = 1$, the answer is simple since $\mathcal{X}$ is approximated by a curve whose equation is $G(\theta_1, x) = 0$, G being the model $G(\theta_1, x) = x - S_{b_1}\left(a_1^T x\right)$ (see (5)). In order to write this model for $d > 1$ in

a concise form, let us denote $a_k^T x = P_{a_k}(x)$, P_{a_k} being the function $\mathbb{R}^n \to \mathbb{R}$ which defines the orthogonal projection of x_j on the axis a_k. With this notation, from the algorithm, we get:

$$\begin{aligned} r_j(1) &= (Id_{\mathbb{R}^n} - S_{b_1} P_{a_1})(x_j) \\ r_j(2) &= (Id_{\mathbb{R}^n} - S_{b_2} P_{a_2})(r_j(1)) \quad ... \\ ... \; r_j(d) &= (Id_{\mathbb{R}^n} - S_{b_d} P_{a_d})(r_j(d-1)). \end{aligned}$$

that can be rewritten as:

$$\begin{aligned} r_j(d) &= (Id_{\mathbb{R}^n} - S_{b_d} P_{a_d})...(Id_{\mathbb{R}^n} - S_{b_2} P_{a_2})(Id_{\mathbb{R}^n} - S_{b_1} P_{a_1})(x_j) \\ &= \left(\coprod_{k=d}^{1} (Id_{\mathbb{R}^n} - S_{b_k} P_{a_k}) \right)(x_j) \; . \end{aligned}$$

Finally, it appears that the representation of $\mathcal{X}$ is defined by $G(\theta, x) = 0$ with:

$$G(\theta, x) = \left(\coprod_{k=d}^{1} (Id_{\mathbb{R}^n} - S_{b_k} P_{a_k}) \right)(x) \; , \tag{14}$$

where $\theta = (\theta_1, ..., \theta_d)$, the error being $r_j(d) = G(\theta, x_j)$.

Properties

The model (14) has the following properties : *(1)* with d axis, $G(\theta, x) = 0$ defines a d-dimensional manifold, *(2)* the errors are decreasing, *(3)* with $d = n$, the model is exact. Proof is given in [17]. Let us note that linear PCA shares these propreties. Like for PCA, we define the information ratio associated to the d-dimensional model:

$$K_d = 1 - \sum_j \|r^j(d)\|^2 / \sum_j \|x^j\|^2 \; .$$

We deduce from the previous proposition that the K_d series is increasing and that $K_n = 1$. K_d allows to choose the model dimension for a given information ratio.

Let us come back to the condition (12). It justifies the residual errors definition. This definition comes directly from the model equation $G(\theta, x) = 0$ by setting $r^j(d) = G(\theta, x^j)$. This way of defining an approximation error may be ill-conditioned. For instance when $G(\theta, x) = 0$ defines a quadratic curve, r_j does not come from an orthogonal projection on the curve (except in the circle case), and r_j can even be infinity for some points close to the curve (see [2]). However, in our case, for a given axis, we avoid this problem thanks to the orthogonality.

4 Experimental results

First experiment

This is an academic example. Fig.1(a) shows a sample of a set of curves obtained by translating a given curve. Let us denote this set as $\{g_j(s), j = 1, ..., N\}$ where $N = 100$ and j corresponds to the translation index. All these curves are sampled on a *same* interval. For each of them, this leads to a vector $x_j = ((g_j(s_1), ..., g_j(s_n))$, with $n = 50$. This set corresponds to a cloud $\mathcal{X}$ in $\mathbb{R}^n$ located on a 1-dimensional manifold. The number of examples N may look small compared to the space dimension n, but the important thing here is to have a large number of samples compared to the intrinsic dimension of $\mathcal{X}$. We can imagine what this manifold looks like by projecting it on the first principal plane computed by PCA (Fig.1(d)). Using the 1-dimensional PCA approximation, we get a straight line. This representation is not sufficient. To get a good approximation of the one-dimensional set $\mathcal{X}$, five axes are necessary. This leads to a five-dimensional linear subspace representing 95% of the cloud variance. PCA simulations provide points in $\mathbb{R}^n$ which are not representative of the training set (Fig.1(b)), illustrating in this case, the very poor PCA generalization ability.

Now, let us consider the NLPCA modeling. We do not know if there is a parametrization axis. The translation, which was used to build the set of examples cannot be written as a linear combination of the coordinates. In order to search for a parametrization axis, the algorithm A' was used for the index maximization. The resulting axis a_1 yields the neighborhood preservation of 93 points among the 100 initial points.
The generalisation criterium ((15) in Appendix) has given an optimal spline smoothing for $\nu = 29$, (Note that the standard cross-validation criterium (16) gives $\nu = 18$). This generalisation criterium was computed using $M = 8000$ simulated values $\{\breve{u}_k,\ k = 1, ..., M\}$ on the axis a_1 , with respect to the probability density of the projection of $\mathcal{X}$ on this axis. It gives M points $S_b(\breve{u}^k)$ in $\mathbb{R}^n$ located on the manifold approximating $\mathcal{X}$. Fig.1(d) shows the projection of theses points on the PCA principal plane.
The corresponding simulated curves are very closed to the ones of $\mathcal{X}$, (Fig.1(d)). Let us note that a bad choice of the number of knots ($\nu = 96$) leads to a very poor generalization behavior (Fig.4).

Second experiment

The object of interest is a lamp for which $N = 45$ appearances have been obtained by varying the azimuth and elevation of the viewpoint [1] (Fig.5). This experiment is closed to that of Murase and Nayar [32] who have shown the nonlinear nature of such data. A one dimensional NLPCA model has been selected. The algorithm A' used with $T_0 = 1$ and $\lambda = 0.995$, has converged after 1000 iterations. It tooks 9 mn on a Pentium 233 Mhz. The axis a_1 keep the neighborhood of 33 points among the 45 points of $\mathcal{X}$. In Fig.6, the

[1] Centre for Intelligent Systems, University of Plymouth
`http://www.cis.plym.ac.uk/cis/3Darchive.html`.

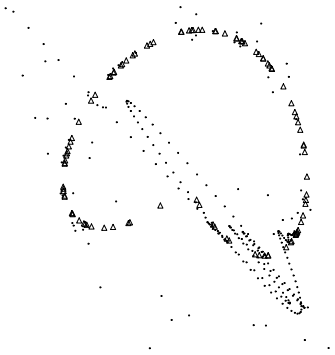

Figure 4: Simulation with a wrong number of knots.

representation of the lamp by NLPCA is more precise than the PCA representation which shows blur contours and detail removing. In particular the knob of the lamp is not depicted.

Third experiment

$\mathcal{X}$ is composed of $N = 400$ face images from 40 individuals [2] (Fig.7). Here, the quality of the representation is measured from the approximation error. To represent faces with 20% of mean error, the PCA model requires $d = 210$ whereas the NLPCA model requires $d = 80$ for $\nu = 1$ and $d = 65$ for $\nu = 2$. In this experiment, let us note that PCA and NLPCA axes are closed. Fig.8 show representation by PCA and NLPCA for $d = 89$ in the two cases. The mean error is 35% for the PCA representation and 20% for the NLPCA one. As for the lamp experiment, we see that the NLPCA model keeps many details contrary to the PCA model which yield severe blurred faces.

5 Conclusion

We proposed a parametric model to approximate a set of points $\mathcal{X}$ which are non-linearly distributed in a large multidimensional space. From the geometric point of view, this representation is done using a d-dimensional manifold, computed by a Projection Pursuit algorithm. The main points of this method are the following : The axes are searched for by optimizing an index which preserves the neighborhood structure of $\mathcal{X}$. When the optimized

[2] Olivetti and Oracle Research Laboratory
`http://www.cam-orl.co.uk/face database.html`.

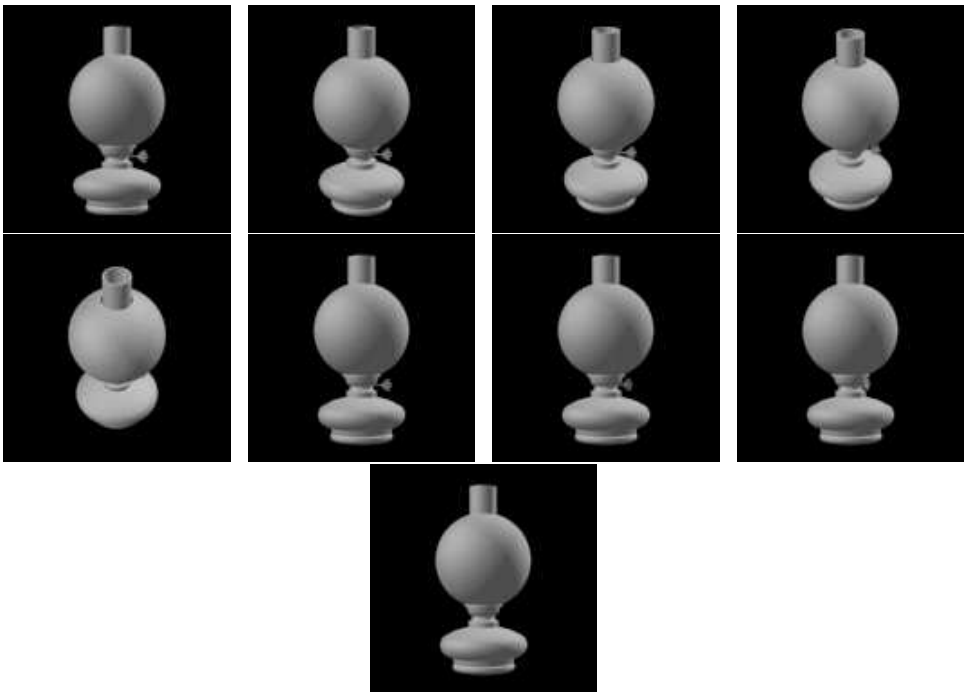

Figure 5: Several appearances of a lamp.

Figure 6: (a) An appearance. (b) PCA representation. (c) NLPCA representation.

Figure 7: Faces.

index is large, the smoothing makes sense because the axis provides a natural order to the points. The second point deals with the ability to control the manifold dimension using the information ratio. Then, the last point is the determination of the smoothing rate using a generalization criterion. These operations are fully automatic, they do not ask the user to adapt some parameters. Finally, when the index and the information values are close to their maximum value, then the model can be considered as valid.

Let us note that the modeling process leads to the estimation of a large number of parameters : axis parameters and spline parameters. However, there is no over-parametrization but on the contrary a dimensionality reduction. This appears clearly when the method is used in a compression framework. Once the model estimated on $\mathcal{X}$, the spline parameters are definitively stocked. A new observation compression x only requires the transmission of d scalars, the projections of x on the d axes. The restoration is given by the points on the manifold relative to the d scalars thanks to the model generalization properties. The compression rate is very important.

In fact, our research has been motivated by the need to synthesis artificial observations x. In this framework, the set $\mathcal{X}$ is seen as an observation set of a flexible "object" and we are more interested by the simulation of deformation than by the approximation. More precisely simulating "object" deformations provides useful *prior* information to solve some inverse problems [15].

Figure 8: PCA and NLPCA representation.

Appendix

Choice of the number of knots. The problem is a classical one, numerous works have dealt with it. A well-known solution is based on the cross-validation error [47]. As it was said above, our final goal is to simulate X in order to get simulations that "looks like" the observations $\mathcal{X}$. With the approximation error alone, the model quality is measured only at the u_j points. The use of the model can lead to unrealistic simulations away from these points, that is to say, very different from the observations $\mathcal{X}$ (see Fig.2(c)). So, we need a criterion which takes into account the smoothing behavior between the points u_j. We propose to use the following generalization error criterion.

Definition 1 *Let U be the random variable X projected on the axis a : $U = a^T X$ and $f(u)$ its probability density on $\mathbb{R}$. The theoretical generalization error is :*

$$\mathcal{G}(\nu) \quad = \quad E[Q^2(\mathcal{X}, S_{\widehat{b}(\nu)}(U))]$$

where Q^2 is the square of a distance, E is the mathematical expectation with respect to $f(u)$ and $\widehat{b}$ is the least-square solution (11) .

In practice, an empirical version $\mathcal{G}^{emp}$ of $\mathcal{G}$ is proposed [45]. Let $\widehat{f}(t)$ be an estimation of the density $f(t)$, (see [40]). In order to define this empirical version, we simulate U according to the $\widehat{f}$ density. Let $\breve{u}$ be such a simulation and $\phi(\breve{u})$ the point u_j which is the closest neighbor of $\breve{u}$ on the axis a : $\phi(\breve{t}) = \arg\min_j (\breve{u} - u_j)^2$. The empirical generalization error is then defined using M realizations $(\breve{u}_1, ..., \breve{u}_M)$ of U as follows :

$$\mathcal{G}^{emp}(\nu) \quad = \quad \frac{1}{M} \sum_{k=1}^{M} \| x_{\phi(\breve{t}_k)} - S_{\widehat{b}(\nu)}(\breve{t}_k) \|^2 \ . \tag{15}$$

$\mathcal{G}^{emp}$ is a quadratic distance between the simulations $\{S_{\widehat{b}(\nu)}(\breve{u}_k)\}$ of X and the initial data $\mathcal{X}$. (At this point, we can see how X is simulated using the random variable U). We choose the number of knots ν that minimizes this criterion. (Let us note that when ν is fixed, the knots are chosen so as to give to each interval defined by two neighboring knots the same probability for $\widehat{f}(t)$). We could easily show that this criterion can be expanded into a variance term and a bias term and that the ν choice realizes a compromise between these two terms [13]: when ν is too large, the bias is small (good data approximation) but the variance is large (bad generalization).
Finally, let us note that the classical cross-validation criterium $\mathcal{C}$ takes also into account the smoothing behaviour between the points u_j by dropping successively one point u_j at a time, as follows:

$$\mathcal{C}(\nu) \quad = \quad \frac{1}{N} \sum_{j=1}^{N} \| x_j - S_{\widehat{b}(\nu,j)}(u_j) \|^2 \ , \tag{16}$$

where $\widehat{b}(\nu, j)$ is the least-square solution based on the data $\mathcal{X} \setminus \{x_j\}$. Yet, when $\mathcal{X}$ suffers a curse of dimensionality [22], this criterium seems to be less adapted than the generalization error criterium (15).

References

[1] J.D. Banfield and A.E. Raftery, "Ice floe indentification in satellite images using mathematical morphology and clustering about principal curves", *Journal of the American Statis. Ass.*, 87, pp. 7-16, 1992.

[2] F.L. Bookstein, *The measurement of biological shape and shape change*, Springer-Verlag, 1978.

[3] J.D. Carroll and P. Arabie, "Multidimensionnal scaling", *Annual Rev. of Psychology*, 31, pp. 607-649, 1980.

[4] N. Chabab, B. Moghaddam and A. Pentland, "Flexible images: Matching and recognition using learned deformations", *Computer Vision and Image Understanding*, 65 (2), pp. 179-191, 1997.

[5] B. Chalmond, *Modelization for Image Analysis*, (in French, to appear), Math. & Appli. Series, Springer-Verlag 1999.

[6] T. Cootes, C. Taylor, D.H. Cooper J. Graham, "Active shape models, their training and application", *Computer Vision and Image Understanding*, 61(1), pp. 38-59, 1995.

[7] P. Diaconis and M. Shashahani, "On nonlinear functions of linear combinations", *SIAM Journal of Scientifical Statis. Comput.*, 5 (1), pp. 175-191, 1984.

[8] T. Duchamp and W. Stuetzle, "Geometric properies of principal curves in the plane", In *Robust Statistics, Data Analysis and Computer Intensive Methods*, H. Rieder (Ed.), Lectures Notes in Statistics, 109, Springer-Verlag, 1995.

[9] M. Duflo, *Stochastic algorithms*, Math. & Appli. Series, Springer-Verlag, 1996.

[10] R.L. Eubank, *Spline smoothing and non-parametric regression*, Decker, 1990.

[11] J.H. Friedman, "Exploratory Projection Pursuit", *Journal of the American Statis. Ass.*, 82 (397), pp. 249-266, 1987.

[12] J.H. Friedman and Stuetzle, "Projection Pursuit Regression", *Journal of the American Statis. Ass.*, 76(376), pp. 817-823, 1981.

[13] S. Geman, E. Bienenstock and R. Dousart, "Neural networks and the bias/variance dilemma", *Neural Computation*, pp. 1-52, 1991.

[14] A. Gifi, *Nonlinear multivariate analysis*, John Wiley and Sons, 1990.

[15] S.C. Girard, J.M. Dinten and B. Chalmond, "Building and training radiographic models for flexible object identification from incomplete data", *IEE Vision, Image Signal Proc.*, 143(4), pp. 257-264, 1996.

[16] S.C. Girard, B. Chalmond and J.M. Dinten, "Designing nonlinear models for flexible curves", *Curves and Surfaces with Applications in CGAD*,

[17] S.C. Girard, "Design and statistical learning for nonlinear auto-associative models", PhD thesis, University of Cergy-Pontoise, 1996, (in French).

[18] S.C. Girard, B. Chalmond and J.M. Dinten, "Position of principal components analysis among auto-associative composite models", *C. R. Académie des Sciences*, Série I, Statistics, 236, pp. 763-768, Elsevier, Paris, 1998.

[19] P.W. Hallinan, "A low-dimensional representation of human faces for arbitrary lighting conditions", *IEEE Conf. Comp. Vis. Patt. Rec.*, pp. 995-999, 1994.

[20] T. Hastie and W. Stuetzle, "Principal Curves", *Journal of the American Statis. Ass.*, 84 (406), pp. 502-516, 1989.

[21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, "Surface reconstruction from unorganized points", *Computer Graphics*, 26, (SIGGRAPH'92 Poc.), pp. 71-78, 1992.

[22] P.J. Huber, "Projection Pursuit", *The Annals of Statistics* 13 (2), pp. 434-525, 1985.

[23] M. Kirby and L. Sirovich, "Application of the Karuhnen-Loève procedure for the characterisation of human faces". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12 (1), pp. 103-108, 1990.

[24] J. Karuhnen and J. Joutsensalo, "Generalization of Principal Component Analysis, optimization problems, and neural networks", *Neural Networks*, 8 (4), pp. 549-562, 1995.

[25] A. Lantinis, C. Taylor and T. Cootes, "Automatic interpretation and coding for face images using flexible models", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (7), pp. 696-710. 1997 .

[26] C. Kervrann and F. Heitz, "Learning structure deformation modes of nonrigid objects in long image sequences", *Proc. Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, 1995.

[27] J.B. Kruskal and M. Wish, *Multidimensional scaling*, Sage, Beverly Hills, Calif. 1978.

[28] K.V. Mardia, J.T. Kent and J.M. Bibby, *Multivariate Analysis*, Academic Press.

[29] S. McKenna, S. Gong and J.J. Collins, "Face tracking and pose representation", *Proc. British Machine Vision Conf.* Edinburg 1996.

[30] J. Milnor, *Topology from the differentiable point of view*, The university press of Virginia, Charlottesville, 1965.

[31] B. Moghaddam and A. Pentland "Probabilistic visual learning for object representation". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (7), pp. 696-710, 1997.

[32] H. Murase and S. Nayar, "Visual learning and recognition of 3D objects from appearance", *International J. of Computer Vision*, 14, pp. 5-24, 1995.

[33] G.M. Nielson, T.A. Foley, B. Hamann and D. Lane, "Visualing and modeling scattered multivariate data", *IEEE CG & A*, 11 (3), pp.47-55, 1991.

[34] V.I. Pavlovic, R. Sharman and T.S. Huang, "Visual interpretation for human-computer interaction: a review", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (7), pp. 677-695, 1997.

[35] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes*, Cambrige University Press, 1986.

[36] J.O. Ramsay and B.W. Silverman, *Functional data analysis*, Springer-Verlag, 1997.

[37] J.W. Sammson, "A nonlinear mapping algorithm for data structure analysis", *IEEE Trans. on Computer*, 18 (5), pp. pp. 401-409, 1969.

[38] E. Saund, "Dimensionality-reduction using connectionist networks", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11 (3), pp. 304-314, 1989.

[39] R.N. Shepard and J.D. Carroll, "Parametric representation of nonlinear data structures", *Int. Symp. on Multivariate Analysis*, P.R. Krishnaiah (Ed.), pp. 561-592, Academic-Press, 1965.

[40] B.W. Silverman, *Density estimation*, Chapman and Hall, 1990.

[41] P.D. Sozou, T.F. Cootes, C.J. Taylor and E.C. Di Mauro, "Nonlinear generalization of point distribution models using polynomial regression", *Image and Vision Computing*, 13 (5), pp. 451-457, 1995.

[42] K.K. Sung and T. Poggio, "Example-based learning for view-based human face detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20 (1), pp. 39-50, 1998.

[43] D.L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18 (8), pp. 831-836, 1996.

[44] M. Uenohara and T. Kanade, "Use of Fourier and Karhunen-Loeve decomposition for fast pattern matching with a large set of templates", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (8), pp.891-898, 1997.

[45] V. Vapnik, *Estimation of dependences based on empirical data*, Springer-Verlag, New-York, 1982.

[46] T. Vetter and T. Poggio, "Linear object classes and image synthesis from single example image", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (7), pp. 733-741, 1997.

[47] G. Whaba, "How to smooth curves and surfaces with splines and cross-validation", In *Proc. 24th Conf. on Design of Experiments* pp. 167-192, US Army Research Office, 1979.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399