Diffusion maps as a framework for shape modeling

Nicolas Thorstensen[a,*], Patrick Étyngier[b], Florent Ségonne, Renaud Keriven[c]

[a] *Computational Science Center, Universität Wien, Austria*
[b] *Philips , Paris, France*
[c] *Imagine Laboratory, Université Paris-Est, Marne-la-Vallée, France*

Abstract

Statistical analysis has proven very successful in the image processing community. Linear methods such as principal component analysis (PCA) measure the degree of correlation in datasets to extract meaningful information from high-dimensional data. PCA was successfully applied in several applications such as image segmentation with shape priors and image denoising. The major assumption in these applications is that the dataspace is a linear space. However, this assumption is mainly wrong and as a consequence several non-linear methods were developed, among which diffusion maps were recently proposed. In this paper we develop a variational framework to compute the pre-image using diffusion maps. The key-problem of *pre-image determination* consists of, given its embedding, recovering a point. Therefore we propose to model the underlying manifold as the set of Karcher means of close sample points. This non-linear interpolation is particularly well-adapted to the case of shapes and images. We then define the pre-image as an interpolation with the targeted embedding. The new methodology can then be used for regularization in image segmentation as well as for shape and image denoising. We demonstrate our method by testing our new non-linear shape prior for shape segmentation of partially occluded objects. Further, we report results on denoising 2D images and 3D shapes and demonstrate the superiority of our pre-image method compared to several state-of-the-art techniques in shape and image denoising based on statistical learning techniques.

Keywords: diffusion maps, pre-image, shape prior, shape denoising, image

[*] Corresponding author

Email addresses: nicolas.thorstensen@univie.ac.at (Nicolas Thorstensen), etyngier@imagine.enpc.fr (Patrick Étyngier), florent.segonne@m4x.org (Florent Ségonne), keriven@imagine.enpc.fr (Renaud Keriven)

segmentation, image denoising.

1. Introduction

An important area of image processing is the development of automated segmentation and denoising algorithms. For example, the segmentation of objects in an image provides a way to study the content of this image. Unfortunately, image segmentation is an ill-posed problem due to various perturbing factors such as noise, occlusions, missing parts, cluttered data, etc. Thus, when dealing with complex images some prior knowledge comes in handy and allows to regularize the segmentation problem. One possibility is to introduce statistical knowledge [23] into the segmentation process. The same can be applied to image denoising [26]. So far, most of these methods have used linear learning algorithms such as Principal Component Analysis (PCA). Linear methods show strong limitations since most measurements do not stem from a linear space as in the case of medical shapes and images in general [12, 21].

Recently, several new non-linear learning algorithms were proposed for analyzing high dimensional data such as characters or image sets. Generally, these complex data sets are governed by only a few intrinsic parameters. A set of images depicting faces for instance, might be governed by three parameters: horizontal rotation, vertical rotation and lighting changes [21]. Such features may be recovered by non-linear dimensionality reduction techniques which are also known under the more common name of *manifold learning*. Manifold learning, the process of extracting the meaningful structure and correcting geometric description present in a set of training points $\Gamma = \{s_0 \cdots s_{m-1}\} \in \mathbb{S}$, has witnessed renewed interest over the past years. These techniques are closely related to the notion of dimensionality reduction, *i.e.* the process of recovering the underlying low dimensional structure or intrinsic parameters of a manifold $\mathcal{M}$ that is embedded in the higher-dimensional space $\mathbb{S}$. Among the most recent and popular techniques are the Locally Linear Embedding (LLE) [32], Isomap [34], Laplacian eigenmaps [2], diffusion maps [7, 16, 21]. As pointed out in [14] several of these techniques are actually kernel methods. Their nonlinearity, as well as their locality-preserving properties, are generally viewed as a major advantage over classical methods such as principal component analysis (PCA) and classical multidimensional scaling [7]. These techniques construct an adjacency graph W_{ij} $(i, j \in 0, \ldots, m-1)$ on the set Γ of training samples that captures the local geometry of Γ - its local connectivity - through the use of a kernel function

$W_{ij} = w(s_i, s_j)$ defined on the training samples. The coefficients of the adjacency matrix W measure the strength of the different edges in the adjacency graph (*i.e.* the similarity between samples). Typically $w(s_i, s_j)$ is a decreasing function of the distance $d_{\mathbb{S}}$ between the training points s_i and s_j. In this work, we use the Gaussian kernel $w(s_i, s_j) = e^{(-d^2_{\mathbb{S}}(s_i,s_j)/2\sigma^2)}$, with σ estimated as the median of all the distances between all training points [1, 21].

1.1. Related Work

Statistical methods for shape processing are very common in computer vision. A seminal work in this direction was published by Leventon et al. [23], adding statistical knowledge into energy based segmentation methods. Their method captures the main modes of variation by performing a PCA on the set of shapes. This was extended to non-linear statistics by Cremers et al. in [9]. The authors introduce non linear shape priors by using a probabilistic version of Kernel PCA (KPCA). Dambreville et.al [10] and Arias et al. [1] developed a method for shape denoising based on Kernel PCA. So did Kwok et al. [20] in the context of image denoising. All methods compute a projection of the noisy datum onto a low dimensional space and compute the pre-image of the projected datum. In [4, 25] the authors propose another kernel method for data denoising, the so called *Laplacian Eigenmaps Latent Variable Model* (LELVM), a probabilistic method. This model provides a dimensionality reduction and reconstruction mapping based on linear combinations of input samples. LELVM performs well on motion capture data but fails on complex shapes (see Fig. 6). Further we would like to mention the work of Pennec [30] and Davis et al. [11], which is related to ours, since it considers the manifold of shapes as a Riemannian manifold to solve the average template matching equation.

1.2. Novelty of our approach

We model a category of shapes[1] as a smooth finite-dimensional sub-manifold of the infinite-dimensional shape space, termed the *shape manifold*. This manifold, which cannot be represented explicitly, is approximated from a collection of shape samples using a recent manifold learning technique called diffusion maps [7, 22]. Diffusion maps generate a mapping, called an embedding, from the original shape space into a low-dimensional space. Advantageously, this mapping

[1] For the sake of simplicity, we will call *shapes our training points, although our approach is general and might be e.g. applied to images*

is an isometry from the original shape space, equipped with a diffusion distance, into a low-dimensional Euclidean space [7] (the reduced space or feature space). In this paper, we exploit the isometrical mapping and the Euclidean nature of the reduced space to design a variational framework for pre-image computation to be used for segmentation with shape priors and denoising. Doing so requires being able to estimate the manifold between training samples and to compute the projection of a shape onto the manifold. Unfortunately, diffusion maps do not give access to such tools. Alternatively, we propose to model the pre-image as a Karcher-mean [17] interpolating between neighboring samples for the diffusion distance. Previous pre-image methods were designed for KPCA. Our motivation for using diffusion maps is derived from the fact that this framework generates a mapping that captures the *intrinsic* geometry of the underlying manifold independently of the sampling. Further, the resulting Nyström extension (see Fig. 1b)) proves to be more "meaningful" far from the manifold and leads to quantitatively-better pre-image estimations. Results on 2D,3D shapes and 2D images are presented and demonstrate the superiority of our method.

Our contribution has four aspects :

(i) we use Nyström extensions [3] which provide a sound and efficient framework for extending embedding coordinates to the full shape or image space of possibly infinite dimension,

(ii) we propose a nearest neighbor search in the reduced space to identify the closest neighbors (in the training set) of any shape in the original infinite dimensional shape space.

(iii) In light of this, we describe a variational framework to solve the pre-image problem and show several applications of it.

(iv) Further, we introduce a new shape prior term for image segmentation through a non-linear energy term designed to attract a shape towards its projection onto the manifold.

The rest of the paper is organized as follows. Section 2 presents the diffusion maps framework and the out-of-sample extension. Section 3 introduces our variational framework for solving the pre-image precisely as well as approximately for the use as shape prior in image segmentation. Numerical experiments on real data are reported in section 4 and the conclusion in section 5.

2. Background

The organizational power of graphs on data sets have been proven to be very powerful within the context of machine learning [2, 3, 7]. The purpose of such algorithms is to extract a meaningful structure of high dimensional data and is also commonly referred to as dimensionality reduction. In these methods the eigenspectrum of the Graph-Laplacian [6] plays a central role. Coifman et al. provide a new motivation for the normalized graph Laplacian by relating it to a random walk on a graph (section 2.1). The interpretation as a random walk matrix on a graph gives rise to a diffusion distance which measures correlation between nodes in a probabilistic manner. Further, the diffusion distance allows to analyze data on multiple scales depending on the number of iterations of the random walk matrix.

2.1. Diffusion maps

We will follow the construction of diffusion maps as described in [7]. The first step in computing diffusion maps is the construction of a symmetric graph where each node s_i corresponds to a data point of $\Gamma = \{s_i\}_{(i=0 \dots m-1)}$. The connectivity between nodes is computed based on some user-defined and application dependent scale parameter σ. Two techniques are encountered in the literature [2] for building the adjacency graph W. First, the ϵ-neighborhood graph which connects all vertices with distance $\|s_i - s_j\|^2$ smaller than ϵ, where the norm is the usual Euclidean norm. This construction directly yields a symmetric graph but often produces graphs with multiple connected components. A common heuristic to build connected ϵ neighborhood graph is to take the mean or median among all entries of W [7]. The second method builds a graph where nodes are connected to node s_i through an edge if they are among the nearest neighbors.

In a second step, we need to weight each edge in the graph in which the weight reflects the similarity between two connected nodes. A common weighting function is the Gaussian kernel $w(s_i, s_j) = exp(-\|s_i - s_j\|^2/(2\sigma^2))$ which is also used in [2, 7]. It is obvious that the distance measure between nodes in the graph depends on the application and must be chosen by the user. But it also shows the generality of this method because we only need to choose a similarity function $w(.,.)$ satisfying the following two conditions: For all s_j and s_i in Γ:

- symmetry $w(s_i, s_j) = w(s_j, s_i)$ and
- non negativity $w(s_i, s_j) \geq 0$.

The weight function provides a notion of similarity and defines the amount of local interaction between nodes. We will elaborate on this idea shortly. The common construction of Graph-Laplacian's [6] starts with the definition of the degree function $d : \Gamma \mapsto \mathbb{R}^{+}$:

$$d(s_i) = \sum_{s_j \sim s_i} w(s_i, s_j) \tag{1}$$

where $\sim$ stands for: s_j adjacent to s_i. Since[2] W is assumed symmetric, $\sim$ is also symmetric. Then the unnormalized Laplacian, also referred to as the combinatorial Laplacian, L_u writes as

$$L_u(s_i, s_j) = \begin{cases} d(s_i), & \text{if } s_i = s_j \\ -w(s_i, s_j), & \text{if } s_i \sim s_j \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

In matrix form L_u writes as

$$L_u = D - W \tag{3}$$

where D is a diagonal matrix so that $D_{i,i} = d(s_i)$ and W is the adjacency matrix containing the weights $W_{ij} = w(s_i, s_j)$. The normalized graph Laplacian L_n is given as

$$L_n(s_i, s_j) = \begin{cases} 1 - \frac{w(s_i,s_j)}{d(s_j)}, & \text{if } s_i = s_j \\ -\frac{w(s_i,s_j)}{\sqrt{d(s_i)d(s_j)}}, & \text{if } s_i \sim s_j \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

Again, we can write this in matrix form as $L_n = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ with I being the identity matrix. In order to clarify the relation between the two Laplacian's, we express the normalized Laplacian in terms of the unnormalized which then writes

$$L_n = D^{-\frac{1}{2}} L_u D^{-\frac{1}{2}}. \tag{5}$$

Diffusion maps [7] uses a diffusion kernel $w(s_i, s_j)$ to build the normalized Laplacian matrix

$$P_{i,j} = \frac{w(s_i, s_j)}{d(s_i)}. \tag{6}$$

The diffusion kernel $w(s_i, s_j)$ encodes the probability of transition between s_i and s_j and $d(s_i)$ normalizes the quantity in equation (6) so that $\sum_{s_j \sim s_i} p(s_i, s_j) = 1$. Therefore, the quantity $p(s_i, s_j)$ can be seen as the probability of a random walker to jump from s_i to s_j and P becomes a Markov Chain on the adjacency graph. If

[2] In practice symmetry is forced by setting $W = \frac{1}{2}(W + W^T)$

we introduce a time index t on the random walk matrix P, where t corresponds to the t-th power of P then the kernel $p^t(s_i, s_j)$ corresponds to the probabilities of transition between s_i and s_j in t time steps with the corresponding affinity matrix is P^t. For a connected graph when $t = \infty$ the random walk converges to a unique stationary distribution ϕ_0. In [2], the author proves that the Gaussian kernel approximates the Laplace-Beltrami operator if the data lies approximately on a Riemannian submanifold with uniform distribution. This idea goes hand in hand with the asymptotic behavior of the diffusion kernel which converges to a constant function and illustrates the well-known averaging behavior of the Laplace-Beltrami operator. Therefore is ϕ_0 an eigenvector of P^t so that $\phi_0^T P^t = \phi_0^T$. Using a well known fact from spectral graph theory, Coifman [7] shows that the matrix P^t is conjugate to a symmetric matrix verifying the following eigendecomposition of the kernel p^t :

$$p^t(s_i, s_j) = \sum_{l \geq 0}^{m-1} \lambda_l^t \psi_l(s_i) \phi_l(s_j). \tag{7}$$

$\{\lambda_i^t\}$ is the decreasing eigenspectrum of P^t and $\{\phi_l(s_j)\}$ respectively $\{\psi_l(s_i)\}$ being the corresponding biorthogonal left and right eigenvectors. Then the diffusion distance $D^t(s_i, s_j)$ between two points s_i and s_j can be written as

$$D_t^2(s_i, s_j) = \sum_{s_l \in \Gamma} \frac{(p^t(s_i, s_l) - p^t(s_j, s_l))^2}{\phi_0(s_l)}. \tag{8}$$

This simple L^2 weighted distance between the conditional probabilities $p^t(s_i, .)$, $p^t(s_j, .)$ defines a metric on the data that measures the amount of connectivity of the points s_i and s_j along paths of length t (c.f. [8] for the proof). Therefore, if the graph has a large number of paths connecting s_i and s_j, then the diffusion distance will be small. This behavior of the diffusion distance is used by Lafon et al. to propose a clustering algorithm using diffusion maps [22]. In order to relate the diffusion distance and the eigenvectors of P^t we insert equation (7) into equation (8) and use the biorthogonality between right- and left eigenvector to find:

$$D_t^2(s_i, s_j) = \sum_{i \geq 0}^{m-1} \lambda_i^t (\psi_i(s_i) - \psi_i(s_j))^2. \tag{9}$$

ψ^t are the right eigenvectors associated to P^t and since ψ_0^t is a constant vector it is left out of the sum. Using equation (9), it can be shown that the right eigenvectors of P^t can be used to compute the diffusion distance. To this end, we introduce the family of diffusion maps indexed by a time parameter t

$$\Psi^t(s_i) = \begin{pmatrix} \lambda_0^t \psi_0(s_i) \\ \lambda_1^t \psi_1(s_i) \\ \vdots \\ \lambda_{m-1}^t \psi_{m-1}(s_i) \end{pmatrix}. \tag{10}$$

In the sequel, we omit the parameter t and assume it is set to a fixed value [21]. From equation (9), we can see that diffusion maps generate a quasi-isometric mapping as the diffusion distance is approximately equal to the L^2 metric in the new coordinate system when retaining the first k eigenvectors. The parameter k is intimately related to the intrinsic dimension of the data. In general, the automated estimation of k is very hard. Usually one looks at the dacey of the eigenvalues to infer the parameter k. See [15] for an algorithm to estimate the intrinsic dimension of submanifolds in $\mathbb{R}^d$. In our method, k is a parameter to be set manually. Also note that methods like LLE or Laplacian Eigenmaps do not provide an explicit metric which is crucial for the contribution of this paper. Let us finally mention that a complete density invariant Markov chain can be build by re-weighting the kernel as:

$$\tilde{w}(s_i, s_j) = \frac{w(s_i, s_j)}{q(s_i)q(s_j)}, \tag{11}$$

with $q(s_i) = \sum_{s_j \in \Gamma} w(s_i, s_j)$. Then the new transition probability becomes

$$p(s_i, s_j) = \frac{\tilde{w}(s_i, s_j)}{\tilde{q}(s_i)} \tag{12}$$

with $\tilde{q}(s_i) = \sum_{s_j \in \Gamma} \tilde{w}(s_i, s_j)$. In this way one can recover the geometry of the manifold independently of the data's density.

2.2. Out-of-sample extension

As we have seen in the previous section, computing diffusion maps involves the computation of eigenvectors of a symmetric matrix of size $m \times m$. This takes a considerable amount of time. Thus the question of how to extend an empirical function outside the range of samples, when a new datum is input, becomes critical. In the machine learning community this problem is usually solved with the Nyström method which numerically approximates eigenfunctions of integral

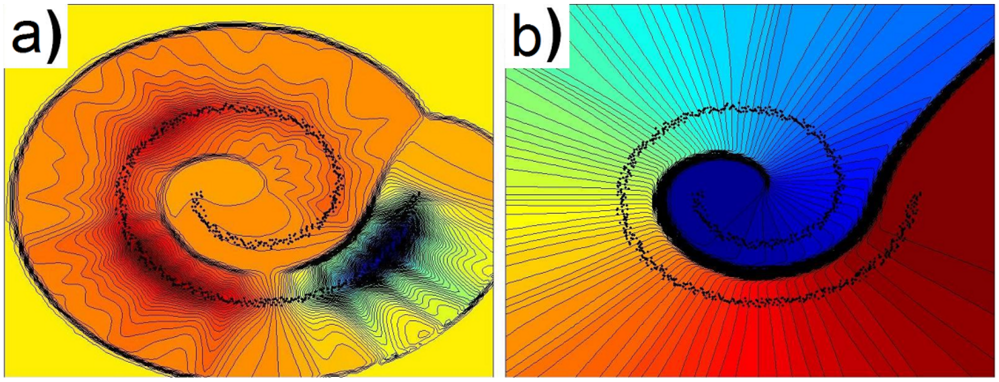

Figure 1: Nyström extension of the first embedding coordinate (from blue corresponding to smaller coordinates to red corresponding to larger ones) provided by the leading eigenvector in different kernel methods, for 1000 points sampling a 2D spiral. From left to right: Kernel PCA (a) and diffusion maps (b).

equations of the form

$$\int_a^b w(x,y)\psi(y)dy = \lambda\psi(x), \tag{13}$$

with $w(.,.)$ being a suitable kernel function. This eigenfunction problem can be approximated by evaluating the equation at evenly space points $\eta_1, \eta_2, \ldots, \eta_n$ on the interval $[a,b]$ with the following quadrature rule

$$\frac{(b-a)}{n}\sum_{j=1}^{n} w(x,\eta_j)\hat{\psi}(\eta_j) = \lambda\hat{\psi}(x). \tag{14}$$

Here $\hat{\psi}(x)$ is an approximation to the true $\psi(x)$. In order to solve equation (14), we set $x = \eta_i$

$$\frac{(b-a)}{n}\sum_{j=1}^{n} w(\eta_i,\eta_j)\hat{\psi}(\eta_j) = \lambda\hat{\psi}(\eta_i). \tag{15}$$

Let $[a,b]$ be$[0,1]$, we rewrite the system of equations in matrix form as

$$W\hat{\Psi} = n\hat{\Psi}\Lambda. \tag{16}$$

Where $W_{ij} = w(\eta_i,\eta_j)$ and $\hat{\Psi} = \left[\hat{\psi}_1, \ldots, \hat{\psi}_n\right]$ are the n eigenvectors of A with associated eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$. Then substituting back into equation (14) yields the desired Nyström extension of x for each $\hat{\psi}_i$

$$\hat{\psi}_i(x) = \frac{1}{n\lambda_i} \sum_{j=1}^{n} w(x, \eta_j)\hat{\psi}_i(\eta_j). \tag{17}$$

This expression allows us to extend an eigenvector computed for a set of sample points to an arbitrary point x in terms of a linear combination of the eigenvectors weighted by the kernel $w(., \eta_j)$.

Nyström and diffusion maps

With this powerful method, we are now able to extend the mapping Ψ outside the training set Γ. Considering that every training sample verifies:

$$\forall s_j \in \Gamma \text{ and } \forall l \in 1, \ldots, m, \quad \sum_{s_i \in \Gamma} p(s_j, s_i)\psi_l(y) = \lambda_l \psi_l(s_i), \tag{18}$$

the embedding of a new datum s' not in the set Γ can similarly be computed by a smooth extension $\hat{\Psi}$ of Ψ :

$$\hat{\Psi} : \left\{ \begin{array}{c} \mathbb{S} \to \mathbb{R}^m, \ s' \mapsto (\hat{\psi}_0(s'), \ldots, \hat{\psi}_{m-1}(s')) \\ \text{where } \hat{\psi}_l(s') = \lambda_l^{-1} \sum_{s \in \Gamma} p(s', s)\psi_l(s) \ (\forall l \in 0, ..., m-1) \end{array} \right\}. \tag{19}$$

With this approximation in mind, we can introduce a variational method for interpolation of shapes and images in diffusion maps.

3. Shape interpolation as Karcher means

We emphasize the manifold interpretation and define the pre-image of $\Upsilon \in \mathbb{R}^k$ as the point $s = \hat{\Psi}^{-1}_{|\mathcal{M}}(\Upsilon)$ in the manifold $\mathcal{M}$ so that $\hat{\Psi}(s) = \Upsilon$ with $s \in \mathbb{S}$. As noted by Arias [1] and Mika [26], the pre-image problem is ill-posed and such a shape might not exist. To circumvent this problem, they search for a pre-image that optimizes a given optimality criterion in the reduced space. This approach was also used in [1, 10].

We are interested in estimating the manifold $\mathcal{M}$ between "neighboring" training samples. Therefore, we assume that the point $s \in \mathbb{S}$ falls inside the convex-hull of the training samples in the reduced space (the space induced by the first

k eigenvector of equation (10) in section 2.1). In this sense, the set of training samples must be exhaustive enough to capture the limits of the manifold $\mathcal{M}$. We then simply exploit the Euclidean nature of the reduced space: for a given Υ, we choose the interpolating subset as its $k+1$ nearest neighbors with respect to the diffusion distance D (c.f. equation (9)) which yields the following optimization problem: Given a set of neighboring points $\mathcal{N} = \{s_1, \cdots, s_{k+1}\}$ (*i.e.* neighboring for the diffusion distance D), we assume that the manifold $\mathcal{M}$ can be locally described (*i.e.* between neighboring samples) by a set of weighted-mean samples $\{s_\Theta\}$ that verifies:

$$\boxed{s_\Theta = \arg\min_{z \in \mathbb{S}} \sum_{1 \leq i \leq k+1} \theta_i d_{\mathbb{S}}(z, s_i)^2, \qquad (20)}$$

where $d_{\mathbb{S}}$ is the distance in the input space and $\left(\theta_i \geq 0,\ \sum_{i=1}^{k+1} \theta_i = 1\right)$. The coefficients $\Theta = \{\theta_1, \ldots, \theta_{k+1}\}$ are the barycentric coefficients of the point s_Θ with respect to its neighbors $\mathcal{N}$ in $\mathbb{S}$. Initially proposed by Charpiat el al.[5], this model has proven to give natural shape interpolations, compared to linear approximations. However, their work is limited to a linearized shape space with small deformation modes around a mean shape

Although diffusion maps extract the global geometry of the training set and define a robust notion of proximity, they cannot permit the estimation of the manifold between training samples, i.e. the local geometry of the manifold is not provided. Following [13, 35], we propose to approximate the manifold as the set of Karcher means [17] interpolating (c.f. equation (20)) between correctly chosen subsets of $k+1$ sample points (k being the dimension of the reduced space). The following sections will justify this choice rather than linear approximations. We propose two ways of choosing the $k+1$ nearest neighbors. The first approach is as in [13], where the authors compute a k-dimensional Delaunay triangulation of the training points in the reduced space. In practice, this limits k to small values because most algorithms available on the Internet compute a Delaunay triangulation in 2D or 3D. Therefore, when $k > 3$, we select the $k+1$ nearest neighbors using [27] as proposed in [35].

```
Input: noisy datum $s'$
Output: denoised datum $\hat{s}$ so
        that $\hat{\Psi}(s') = \hat{\Psi}(\hat{s})$
compute the $k$ nearest
neighbors $\mathcal{N}$;
while $\|\hat{\Psi}(s) - \Psi(s')\| >$error
do
    1) compute $\nabla_s E_\psi$ (equation
    (21));
    2) compute the tangent space
    $\mathbb{T}^{\mathcal{M}}_{s_\Theta}$ (equation (22)) at current
    $s_\Theta$;
    3) project $\nabla_s E_\psi$ onto $\mathbb{T}^{\mathcal{M}}_{s_\Theta}$;
    4) integrate projected
    gradient;
    5) compute the extension
    $\hat{\Psi}(s)$ for the current estimate;
end
```

Algorithm 1: Interpolation constrained to remain a Karcher mean

```
Input: noisy datum $s'$
Output: denoised datum $\hat{s}$ so
        that $\hat{\Psi}(s') = \hat{\Psi}(\hat{s})$
compute the $k$ nearest
neighbors $\mathcal{N}$;
while
$\sum_{s_i \in \mathcal{N}} \theta_i d^2(s, s_i) >$error do
    1) compute $\nabla_s E_\psi$ (equation
    (26));
    2) compute the tangent space
    $\mathbb{C}_s^\perp$;
    3) project gradient $\nabla_s E_\psi$
    onto $\mathbb{C}_s^\perp$;
    4) integrate projected
    gradient;
    5) compute the extension
    $\hat{\Psi}(s)$ for the current estimate;
end
```

Algorithm 2: Interpolation with constant embedding constraint.

3.1. Pre-Image as manifold interpolation

We propose to define the pre-image of a target point Υ in the reduced space, as the point s_Θ that minimizes the energy $E_\Psi(s_\Theta) = \|\hat{\Psi}(s_\Theta) - \Upsilon\|^2$, s_Θ being expressed by a Karcher mean for the neighborhood $\mathcal{N}$ made of the $k+1$ samples of Γ which embedded are the $k+1$-closest neighbors of Υ in the reduced space equipped with D:

$$\boxed{\begin{aligned} \Psi^{-1}_{|\mathcal{M}}(\Upsilon) &= \arg\min_{s_\Theta} \|\hat{\Psi}(s_\Theta) - \Upsilon\|^2, \\ \text{where } s_\Theta &= \arg\min_{z \in \mathbb{S}} \sum_{1 \le i \le k+1} \theta_i d_{\mathbb{S}}(z, s_i)^2 \end{aligned}} \tag{21}$$

When the input space is a certain Euclidean space $\mathbb{R}^n$ with its traditional L^2-norm, this indeed amounts to assuming that the manifold $\mathcal{M}$ is piecewise-linear (*i.e.* linearly interpolated between neighboring training samples). For shapes, we will see that this yields natural pre-images.By simple extension, we define the projection of any new test sample s on the manifold $\mathcal{M}$ by $\Pi_{\mathcal{M}}(s) = \hat{\Psi}^{-1}_{|\mathcal{M}}(\hat{\Psi}(s))$. As a first approach to solve for pre-image we propose to interpolate the solution starting from a known sample. Therefore we compute the pre-image $\hat{\Psi}^{-1}_{|\mathcal{M}}(\Upsilon)$ by gradient descent. Instead of optimizing over Θ, we use a descent over s_Θ itself (equation (21)), so that s_Θ remains a Karcher mean (equation (20)). This boils down to iterated projections of the deformation field $\nabla_s E_\psi$ onto the tangent space $\mathbb{T}^{\mathcal{M}}_{s_\Theta}$ of $\mathcal{M}$ at point s_Θ. Note that to compute the tangent space, we are implicitly assuming that the space $\mathbb{S}$ has a manifold structure, in particular that the tangent space $\mathbb{T}^{\mathbb{S}}_{s_\Theta}$ of $\mathbb{S}$ at location s_Θ (*i.e.* the space of local deformations around s_Θ) is equipped with an inner product that we denote $\langle . | . \rangle_{\mathbb{S}}$.

The optimality condition of equation (20) is:

$$\forall \vec{\beta} \in \mathbb{T}^{\mathbb{S}}_{s_\Theta}, \ \sum_{i=1}^{k+1} \theta_i d_i \langle \nabla_s d_i | \vec{\beta} \rangle_{\mathbb{S}} = 0, \tag{22}$$

where we denote $\mathcal{N} = \{s_1, ..., s_{k+1}\}$ and $d_i = d_{\mathbb{S}}(s_\Theta, s_i)$. In order to recover the tangent space $\mathbb{T}^{\mathcal{M}}_{s_\Theta}$ at s_Θ, one needs to relate the k-independent modes of variations of the coefficient Θ (remember that $\sum_{i=1}^{k+1} \theta_i = 1$) with local deformation fields $\vec{ds}_\Theta \in \mathbb{T}^{\mathbb{S}}_{s_\Theta}$. To a small variation of the barycentric coefficients $\Theta \to \Theta + \vec{d\Theta}$, corresponds a small deformation of the sample $s_\Theta \to s_\Theta + \vec{ds}_\Theta$. Differentiating the optimality condition with respect to Θ and s_Θ provides the relation between $\vec{d\Theta}$ and $\vec{ds}_\Theta$. For example, when the input space is taken to be the Euclidean space, *i.e.* $\mathbb{S} = \mathbb{R}^n$, we obviously obtain $\vec{ds}_\Theta = \sum_1^{k+1} d\theta_i s_i$. Remembering $\sum_1^{k+1} d\theta_i = 0$ and fixing the $d\theta_i$ appropriately, we can recover $\mathbb{T}^{\mathcal{M}}_{s_\Theta}$. Therefore we optimize for s_Θ without explicitly computing Θ. The gradient descent generates a family of samples $s : \tau \in \mathbb{R}^+ \mapsto s(\tau) \in \mathcal{M}$ so that

$$s(0) = s_0, \ \frac{ds}{d\tau} = -\vec{v}_{\mathcal{M}}(s_\tau), \tag{23}$$

with $s_0 \in \mathcal{N}$ (in practice, the nearest neighbor of Υ). The velocity field $\vec{v}_{\mathcal{M}}(s_\tau)$ is the orthogonal projection of the deformation field

$$\nabla_{s_\tau} E_\Psi = (\hat{\Psi}(s_\tau) - \Upsilon)^T \Lambda \Psi^T \nabla_{s_\tau} p_{s_\tau} \tag{24}$$

onto the tangent space $\mathbb{T}^{\mathcal{M}}_{s_\tau}$. Here Λ is a diagonal matrix of eigenvalues and Ψ are the corresponding eigenvectors. Note that before projecting onto $\mathbb{T}^{\mathcal{M}}_{s_\tau}$ we first orthogonalize the tangent space by using Gram-Schmidt. In the case of the L^2-norm, the Θ's can be easily recovered. When using a different distance function such as the symmetric difference or the Sobolev $W^{1,2}$-norm, one needs to solve an additional system of linear equations in each step of the gradient descent (see end of this section for shape metrics).

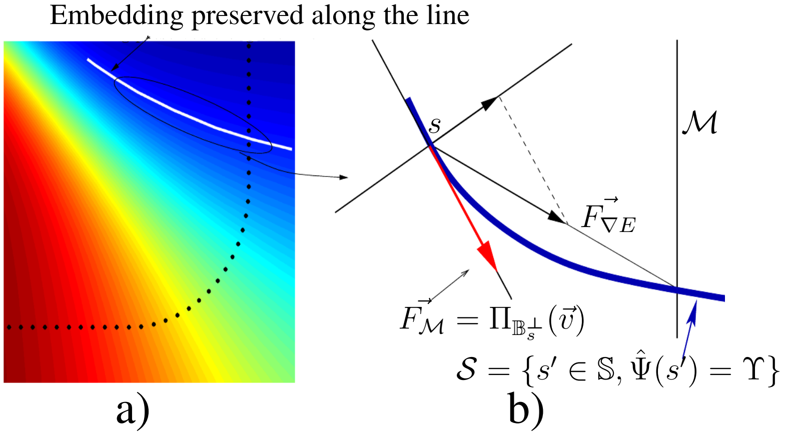
Embedding preserved along the line
s
$\mathcal{M}$
$F_{\vec{\nabla E}}$
$\vec{F_{\mathcal{M}}} = \Pi_{\mathbb{B}_s^\perp}(\vec{v})$
$\mathcal{S} = \{s' \in \mathbb{S}, \hat{\Psi}(s') = \Upsilon\}$
a)
b)

Figure 2: Evolution towards the manifold that preserves the embedding. In a), we see a plot of the isovalues of the extended eigenfunctions. b) illustrates the projection of the gradient $(F_{\vec{\nabla E}})$ (see equation (26)) onto the space $\mathbb{B}_s^\perp$ (see section 3.2). This yields a new force $F_{\mathcal{M}}$ attracting the shape towards $\mathcal{M}$ at constant embedding.

3.2. Pre-image with constant embedding

In this section, we develop a different way of computing the pre-image which is well suited for the use of segmentation with shape priors. The general idea in this section is to minimize equation (20) under the constraint that the computed extension coordinates remain constant during the evolution. In Fig. 2a), we represent the embedding values using the Nyström extension, "around" the 1D swiss roll manifold. We see that attracting the point s towards $\mathcal{M}$ while keeping a constant embedding is a natural idea. Minimization of equation (20) by unconstrained gradient descent would alter the embedding coordinates. Therefore, we propose an optimization procedure that preserves the embedding along the entire evolution path. The difference to the previous optimization is that here we move along the isolevel set of the out of sample extension. Instead of projecting the gradient flow onto the tangent space of the manifold, we project the deformation field onto the tangent space of the isolevel set in the reduced space. For instance, let $\mathcal{S}$ be the iso level set of constant embedding points around s' (see Fig.2b))

$$\mathcal{S} = \{s' \in \mathbb{S}, \hat{\Psi}(s') = \Upsilon\}. \tag{25}$$

The problem is, given Υ, we evolve s on $\mathcal{S}$ towards $\mathcal{M}$. To do so, we choose $\Pi_{\mathcal{M}}(s)$ as a target point and evolve s towards $\Pi_{\mathcal{M}}(s)$ at constant embedding.

Using equation (19), we define the projection of any shape s on the manifold $\mathcal{M}$ by $\Pi_{\mathcal{M}}(s) = \hat{\Psi}^{-1}_{|\mathcal{M}}(\hat{\Psi}(s))$. To projection is computed by minimizing the following energy functional : We define the energy $E_{\mathcal{N},\Theta}$ by the following functional:

$$\boxed{s \mapsto E^{sp}_{\mathcal{N},\Theta}(s) = \sum_{s_i \in \mathcal{N}} \theta_i d^2(s, s_i), \tag{26}}$$

where the coefficients $\Theta = \{\theta_0, \dots, \theta_k\}$ are the solutions to equation (20) with $x = \hat{\Psi}(s)$. Minimization of $E_{\mathcal{N},\Theta}$ by gradient flow produces an evolution which deforms the shape s towards its projection onto the manifold $\Pi_{\mathcal{M}}(s_0)$. Yet, the embedding coordinates $\hat{\Psi}(s(\tau))$ of the evolving shape $s(\tau)$ are not guaranteed to remain constant during the evolution. To counteract this problem, we project the gradient flow $\nabla E_{\mathcal{N},\Theta}$ onto the tangent space of $\mathbb{T}_{\hat{\Psi}(s(\tau))}$. $\mathbb{S}$ is the ambient space (the shape space in our case) that embeds the learned manifold $\mathcal{M}$. Each point $s \in \mathbb{S}$ has its associated deformation space, the tangent space denoted $\mathbb{T}_s$. For instance, when $\mathbb{S}$ is the shape space, $\mathbb{T}_s$ corresponds to normal deformation fields that can be applied to shape s. We define $\mathbb{B}_s$ the space spanned by $\vec{b_i} = \nabla\hat{\Psi}_i(s)$ $(\forall i \in \{1, \cdots, k\})$. The space $\mathbb{B}_s$ is intuitively the space of deformations at s that maximally modifies the embedding. On the opposite side, the space denoted $\mathbb{B}_s^{\perp}$, orthogonal to $\mathbb{B}_s$, corresponds to the deformations that have a minimal influence (tangential deformation field) on the value of the embedding. We then write the deformation space at s as the direct sum between $\mathbb{B}_s$ and $\mathbb{B}_s^{\perp}$:

$$\mathbb{T}_s = \mathbb{B}_s \oplus \mathbb{B}_s^{\perp} \tag{27}$$

We finally calculate an orthonormal basis $\mathbb{C} = \{\vec{c}_1, \cdots, \vec{c}_k\}$ out of $\mathbb{B}_s^{\perp}$ using the orthogonalization Gram-Schmidt process. In order to preserve the embedding during the evolution, we define the projection of any velocity field $\vec{v}$ onto the space $\mathbb{C}_s^{\perp}$:

$$\Pi_{\mathbb{C}_s^{\perp}}(\vec{v}) = \vec{v} - \sum_{i=1}^{k} \langle \vec{v}, \vec{c}_i \rangle \vec{c}_i. \tag{28}$$

As in Fig.2b), we define the force $F_{\mathcal{M}}$ that attracts a point $s \notin \mathcal{M}$ towards the projection $\Pi_{\mathcal{M}}(s)$ so that the embedding is preserved. In summary, the procedure relies on the out-of-sample extension equation (19) and the fact that the reweighted kernel equation (12), $\mathbb{T}_{\hat{\Psi}(s)}$ can be expressed by k simple orthogonality conditions in the tangent space $\mathbb{T}_{\mathbb{S}}(s_\tau)$ of $\mathbb{S}$ at s_τ:

$$\mathbb{T}_{\hat{\Psi}(s)} = \left\{ \begin{array}{c} \overrightarrow{v} \in \mathbb{T}_{\mathbb{S}}(s) \text{ such that } \forall l = 1, \ldots, k \\ \sum_{y \in \Gamma} \langle \nabla_s p(s,y) | \overrightarrow{v} \rangle_{\mathbb{L}^2} \Psi_l(y) = 0 \end{array} \right\}, \tag{29}$$

where $\langle . | . \rangle_{\mathbb{L}^2}$ corresponds to the $\mathbb{L}^2$-dot product in the tangent shape space $\mathbb{T}_{\mathbb{S}}(s)$. Projection of the velocity field $-\nabla E_{\mathcal{N},\Theta}$ onto $\mathbb{T}_{\hat{\Psi}(s)}$ by using equation (28).

3.3. Shape metrics

In order to make use of diffusion maps, we need to define a similarity measure for shapes. One classical choice is the area of the symmetric difference between the regions bounded by the two shapes:

$$d_{sd}(s_1, s_2) = \frac{1}{2} \int |\chi_{\Omega_1} - \chi_{\Omega_2}| \ , \tag{30}$$

where χ_{Ω_i} is the characteristic function of the interior of shape s_i. This distance was recently advocated by Solem in [33] to build geodesic paths between shapes, however, it yields no unique geodesics. Another definition has been proposed [5, 23, 31], based on the representation of a curve in the plane, of a surface in 3D space, by its signed distance function. In this context, the distance between two shapes can be defined as the L^2-norm or the Sobolev $W^{1,2}$-norm of the difference between their signed distance functions. Let us recall that $W^{1,2}(\Omega)$ is the space of square integrable functions over the domain Ω with square integrable derivatives:

$$d_{L^2}(s_1, s_2)^2 = ||\mathbb{D}_{s_1} - \mathbb{D}_{s_2}||^2_{L^2(\Omega,\mathbb{R})} \ , \tag{31}$$

$$d_{W^{1,2}}(s_1, s_2)^2 = ||\mathbb{D}_{s_1} - \mathbb{D}_{s_2}||^2_{L^2(\Omega,\mathbb{R})} + ||\nabla\mathbb{D}_{s_1} - \nabla\mathbb{D}_{s_2}||^2_{L^2(\Omega,\mathbb{R}^n)} \ , \tag{32}$$

where $\mathbb{D}_{s_i}$ denotes the signed distance function of shape s_i ($i = 1, 2$), and $\nabla\mathbb{D}_{s_i}$ being its gradient.

4. Experimental Results

Our experimental section runs on several examples using synthetic and real data. It is organized around the two optimization problems presented in the previous section. First, we validate the purpose of the projection of the gradient onto the tangent space. Then, we test the Karcher mean interpolation on the reconstruction problem of occluded 3D medical shapes as presented in [10]. A last experiment demonstrates the superiority of our method for a standard denoising problem on images. For the second optimization problem we applied the same testing protocol. First, we perform a proof of concept and show how it can be applied to the problem of denoising. When comparing to previous method, we used the parameter settings as described in the original papers. Then, we focus on image segmentation incorporating statistical information.

We work on two types of data. The first type are images and the second type are shapes. In this work images are represented as vectors by concatenating the columns of an image into a vector. When dealing with shapes, we will use a level set representation of the shape. For a good introduction on the numerical implementation and calculus with the level set method please refer to [29]. When dealing with shapes (or distance functions), we use a level set implementation for the gradient flow where the evolution reads as : $S_t = \mathcal{N}\mathcal{F}$. $\mathcal{F}$ weights the normal flow according to the image content and, in some of our experiments, and some a priori knowledge.

Please note that we don't report any timing measures for our method as the computational effort involved to compute a solution is neglectable. And by neglectable we mean fractions of seconds. This is due to the local nature of our optimization problems.

4.1. Remaining on the manifold

To validate both the Karcher means modeling of the manifold and our projecting constraint (Section 3.1), we generate a set of 200 synthetic shapes (or distance functions) parameterized by an articulation angle and a scaling parameter (Fig. 3a). The corresponding embeddings are shown Fig. 3b). Choosing two distant shapes A and B, we compute a path $s(\tau)$ from A to B by mean of a gradient descent starting from $s(0) = A$ and minimizing $d_{\mathbb{S}}(s(\tau), B)$. Fig. 3c) and 3b) show the intermediate shapes and the corresponding embeddings in red. In purple the same path is shown when projecting the gradient in order to remain on the manifold. Observe how, in that case, the intermediate shapes look more like the original sample ones. Note also, that when remaining on $\mathcal{M}$, the interpolating path is almost a straight line with respect to the diffusion distance. The implementation is done using a level set implementation where the force $\mathcal{F}$ is computed from equation (22). The overall optimization is done using Algorithm (1). The gradient in each iteration is weighted with 0.6.

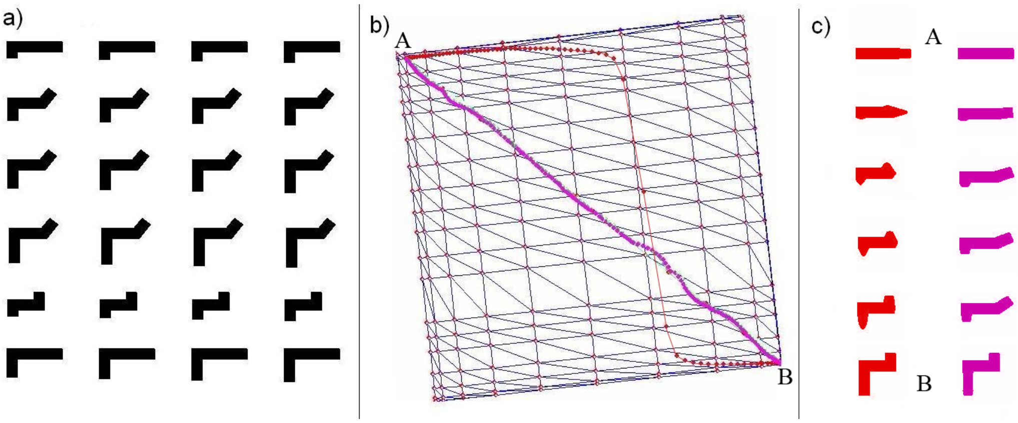

Figure 3: Synthetic sample of 200 articulated and elongated shapes. From left to right: (a) a subset of the sample. (b) triangulated 2-dimensional embedding computed using diffusion maps and a gradient descent from an initial shape to a target one, without (red dots) and with (purple dots) remaining on the interpolated manifold. (c) Some shapes of the resulting evolution (left column: without projection, right column: with projection.

4.2. Maintaining the embedding constant

We now use a dataset of 39 ventricles nuclei from Magnetic Resonance Image (MRI). The shapes are aligned using their principal moment before computing their diffusion coordinates. In this experiment, we compare the projection at constant embedding, the neighbors in the Delaunay triangulation of the reduced space and the mean shape obtained from these neighbors. Our deformation surface is

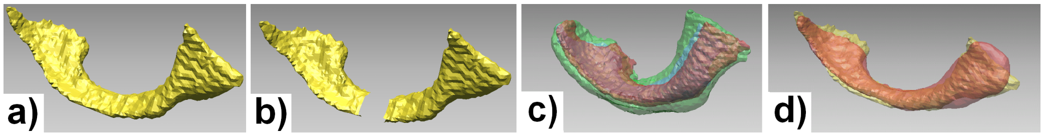

Figure 4: Interpolation using Karcher means for 39 three-dimensional sample shapes. From left to right: a) a new shape not in the given sample b) the same shape with an occlusion c) the 3 nearest neighbors of the corrupted shape according to the diffusion distance (in red, green and blue) d) the original shape (in yellow) and our interpolation (in red). See Table 1 for quantitative results.

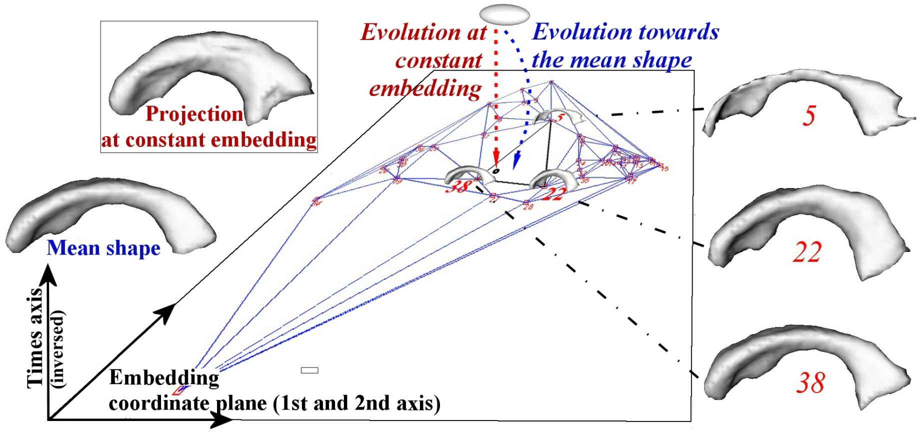
Evolution at constant embedding
Evolution towards the mean shape
Projection at constant embedding
Mean shape
Times axis (inversed)
Embedding coordinate plane (1st and 2nd axis)
5
22
38

Figure 5: The ventricle manifold: Comparison of the evolution towards the mean shape and the evolution at constant embedding

.

again implemented in the level set framework: the distance functions of the ventricle shapes are encoded in 140 × 75 × 60 images. To perform the projection, we start from an ellipsoid aligned on the 3D shape set. Its embedding is indicated by the black point in Fig. 5. The nearest shapes in the corresponding Delaunay triangle are easily identified in order to compute the mean shape target and the projection at constant embedding. The projection at constant embedding captures details (on the right side of the ventricle) of closest shapes (38 & 22) that the mean shape lose due to its smoothing properties. Again, we implemented this method using a level set representation where the force term $\mathcal{F}$ is derived from equation (24). Following Algorithm (2), we perform a weighted gradient descent with weight a constant weight 0.65.

4.3. Projection and manifold as Karcher means

Here we test the validity of using Karcher means as a manifold interpolation model.We consider the space of two-dimensional surfaces embedded in $\mathbb{R}^3$. For such a general space, many different definitions of the distance between two shapes have been proposed in the computer vision literature but, there is no agreement on the correct way to measure shape similarity. In this work, we represent a surface s_i in the Euclidean embedding space $\mathbb{R}^3$ by its signed distance function $\mathbb{D}_{s_i}$. In this context, we define the distance between two shapes to be the L^2-norm of the difference between their signed distance functions [23]:

$$d_{\mathbb{S}}(s_1, s_2)^2 \;=\; ||\mathbb{D}_{s_1} - \mathbb{D}_{s_2}||^2_{L^2}$$

Note that, in order to define a distance between shapes that is invariant to rigid displacements (*e.g.* rotations and translations), we first align the shapes using their principal moments before computing distances. Note also that the proposed method is obviously not limited to a specific choice of distance [5, 13]. We use a dataset of 39 ventricles nuclei extracted from Magnetic Resonance Image (MRI). We learn a random subset of 30 shapes and corrupt the nine remaining shapes by an occlusion (Fig.4a,b). In order to recover the original shapes with our method, we project the shapes onto the shape manifold with our method. We then compare the reconstruction results with the nearest neighbor, the mean of the m+1 nearest neighbors and the method of Dambreville [10]. The parameters of this experiment are $k = 2$. In Fig. 4d) one example of a reconstructed shape (red) is obtained from the $k + 1$ nearest neighbors of $s_\bullet$ (Fig. 4c)). In order to quantitatively evaluate the projection, we define the reconstruction error as $e(s) = d_{\mathbb{S}}(s_\circ, s)/\sigma$, where $s_\circ$ is the original shape and s is the reconstructed shape. The occluded shape has an error of $e(s_\bullet) = 4.35$, while the nearest-neighbor has an error of 1.81. In table 1 we see that our method is superior to the one proposed by Dambreville et.al. [10].

Average error of shapes with occlusion 4.67	Nearest neighbors(NN) 1.81	Mean of NN 1.96
Dambreville et al. 1.1	Our method 0.58	/ /

Table 1: Average reconstruction error for a set of 9 noisy shapes

4.4. Denoising of Digits

To test the performance of our approach on the task of image denoising, we apply the algorithm on the USPS dataset of handwritten digits[3]. In a first experiment, we compare our method to five state-of-the-art algorithms [10], [10]+[1], [20], [20]+[1] and [4]. For each of the ten digits, we form two training sets composed of randomly selected samples (60 and 200 respectively). The test set is composed of 40 images randomly selected and corrupted by some additive Gaussian noise at different noise levels. The process of denoising simply amounts to

[3] The USPS dataset is available from http://www.kernel-machines.org.

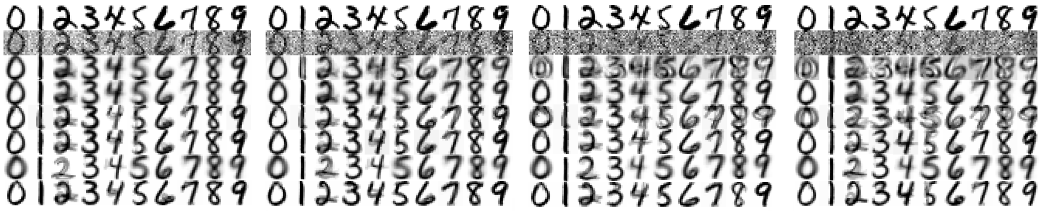

Figure 6: Digit images corrupted by additive Gaussian noise (from left to right, $\sigma^2 = 0.25, 0.45, 0.65, 0.85$). The different rows respectively represent, from top to bottom: the original digits; the corrupted digits; denoising with [10] ; with [10]+[1] ; with [20] ; with [20]+[1] ; with [4]; with our Karcher mean interpolation based method. See Table 2 for quantified results

estimating the pre-images of the feature vectors given by the Nyström extension of the noisy samples. For all the methods, we take $k = 8$ for the reduced dimension (number of eigenvectors for the kernel-PCA based methods). Table 2 shows a quantitative comparison based on the pixel-signal-to-noise ratio (PSNR). Our method visually (Fig. 6)) and quantitatively outperforms other approaches. Interestingly, it is less sensitive to noise than other ones and yields good results even under heavy noise.

4.5. Image segmentation

One way to segment an image into foreground (the object of interest) and background is to define an energy

$$E(\mathcal{C}) = E_{external}(\mathcal{C}, \mathcal{I}) + E_{internal}(\mathcal{C}) \tag{33}$$

depending on the curve $\mathcal{C}$ (the boundary between foreground and background). This objective function has two terms. $E_{external}$ measures how well the current curve separates the two regions and $E_{internal}$ measures the smoothness of the curve and is a regularization term. A concrete model for computation is the active contour model[18]. When used to minimize the equation (33), we have the following minimzation problem

$$E(\mathcal{C}, \mathcal{I}) = \int_0^{L(\mathcal{C})} \left\| \frac{\partial \mathcal{C}}{\partial s} \right\|^2 ds + \int_0^{L(\mathcal{C})} g(\|\nabla \mathcal{I}(\mathcal{C}(s)))\|) ds. \tag{34}$$

$L(\mathcal{C})$ is the length of the curve. Minimizing the previous equation is equivalent to solving a length minimizing curve of

$$E(\mathcal{C}) = \int_0^{L} g(\|\nabla \mathcal{I}(\mathcal{C}(s)))\|) ds. \tag{35}$$

σ^2	Dambreville et al.	Kwok and Tsang	Carreira-Perpiñan and Lu
0.25	8.50	15.71	14.01
0.45	9.05	13.87	13.91
0.65	9.78	13.10	13.89
0.85	9.06	12.58	13.87
0.25	9.35	16.08	15.27
0.45	9.64	15.70	14.85
0.65	9.41	13.97	14.13
0.85	9.24	13.06	14.07
σ^2	Arias et al. +Dambreville et al.	Arias et al. +Kwok and Tsang	Our method
0.25	10.17	16.18	17.71
0.45	9.98	15.42	17.52
0.65	9.58	13.60	17.38
0.85	8.61	13.91	17.32
0.25	11.97	16.21	17.95
0.45	10.18	15.98	17.85
0.65	10.26	15.85	17.79
0.85	10.25	15.07	17.75

Table 2: Average PSNR (in dB) of the denoised images corrupted by different noise levels σ. Training sets consist of 60 samples (first 4 rows) and of 200 samples (last 4 rows).

on a Riemannian space with the metric defined using the image function. A local minimum to equation (35)can be obtained by gradient descent. When taking only the normal component of the gradient, the curve evolution is

$$\frac{\partial \mathcal{C}}{\partial \tau}(s) = F(\mathcal{K}, \mathcal{C}(s), \mathcal{I})\mathcal{N} \tag{36}$$

$$\mathcal{C}(s, 0) = \mathcal{C}_0(s), \tag{37}$$

where F controls the speed of the normal flow. The speed depends on the contour mean curvature and the image data. $\mathcal{N}$ is the normal to the curve. When the shape is represented as the zero level set of some distance function $\mathbb{D}$, then we can write the evolution equivalently [19]

$$\frac{\partial \mathbb{D}(x, y)}{\partial \tau} = F(\mathcal{K}(x, y), \mathcal{I}(x, y))|\nabla \mathbb{D}(x, y)| \Leftrightarrow \frac{\partial \mathcal{C}}{\partial \tau}(s) = F(\mathcal{K}, \mathcal{C}(s), \mathcal{I})\mathcal{N} \tag{38}$$

where $\mathcal{K}(x,y) = \nabla\left(\frac{\nabla\mathbb{D}(x,y)}{|\nabla\mathbb{D}(x,y)|}\right)$. The evolution of the implicit representation, on the left hand side of equation (38), is computed pointwise over the domain. From the previous equation, we see that the evolution is driven by the speed term $\mathcal{F}$. Model-based segmentation incorporates a priori knowledge on the shape to segment. This can be done by extending the model in equation (33) by a third term :

$$E(\mathcal{C}) = E_{external}(\mathcal{C},\mathcal{I}) + E_{internal}(\mathcal{C}) + E_{shape}(\mathcal{C}). \tag{39}$$

4.5.1. Ventricle nucleus segmentation from MRI with occlusion

We consider a simple segmentation task which consists of segmenting the ventricle nucleus from an MRI that was corrupted by white noise and degraded with an artificial occlusion (clearly visible in Fig. 7a). Motivated by our choice of representing a shape s by its signed distance function $\mathbb{D}_s$, our surface deformation is implemented in the level set framework. The level set evolution is guided by a *simple* intensity-based velocity term[23], a curvature term κ, and the non-linear shape prior term $v_{\mathcal{M}}$

$$\begin{aligned}\frac{\partial\mathbb{D}_s(x,y,z)}{\partial\tau} &= [\beta(I(x,y,z) - T(x,y,z)) - \kappa(x,y,z)]|\nabla\mathbb{D}(x,y,z)| \\ &\quad - \alpha v_{\mathcal{M}}(x,y,z)\cdot\mathbb{D}(x,y,z)\end{aligned} \tag{40}$$

where $I(x,y,z)$ and κ represent respectively the image intensity and mean curvature at location (x,y,z), T is a threshold (c.f.[23]) computed locally from image intensities, $\beta = 0.1$ and $\alpha = 0.1$ are two weigthing coefficients. Figure 7 displays our segmentation results. Despite the artificial occlusion, the shape prior term was able to recover the correct shape by attracting the shape onto the *shape prior manifold*. Yet, the final surface is geometrically accurate because the active contour can evolve freely inside the manifold $\mathcal{M}$ subject to the image term. Finally, note that, in practice, the shape prior term is not used during the first steps of the evolution (a robust alignment being impossible).

4.5.2. Segmentation of 2D Cars

In this example, we illustrate the shape prior term in segmentation tasks of 2D car shapes. We are aiming at segmenting partly occluded cars. In this experiment, the non-linear prior is the manifold of the 2D shapes observed while turning around different cars. The used dataset is made up of 17 cars whose shapes are quite different : Audi A3, Audi TT, BMW Z4, Citroën C3, Chrysler Sebring, Honda Civic, Renault Clio, Delorean DMC-12, Ford Mustang Coupe, Lincoln MKZ, Mercedes S-Class, Lada Oka, Fiat Palio, Nissan 200sx, Nissan Primera,

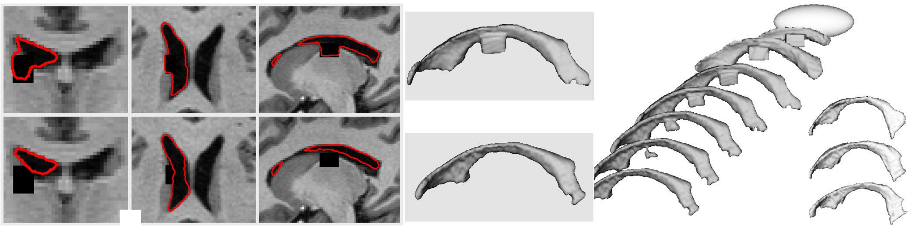

Figure 7: a) Coronal, horizontal, and sagittal slices of the MRI volume with the final segmentations without (top) and with (bottom) the shape prior. b) Some snapshots of the shape evolution - the shape prior term was not used during the first steps. c) The closest neighbors of the final surface.

Hyundai Santa Fe and Subaru Forester. For each car, we extracted 12 shapes from the projection of the 3D CAD model (Fig. 8a) forming a dataset of 204 shape samples . The shapes are finally stored in the form of distance functions by means of 160×120 images. In the learning stage, the embedding of the *car shape manifold* is estimated using diffusion maps over the dataset. In Fig. 8b), we represented the two first dimensions of the diffusion coordinates, which represents the reduced space, and the corresponding Delaunay triangulation. Note that the car shapes have a coherent spatial organization in the reduced space. Actually, diffusion maps provide a parameterization of the car manifold [21]. In this example, the manifold is parameterized by the viewing angles. For the segmentation, we

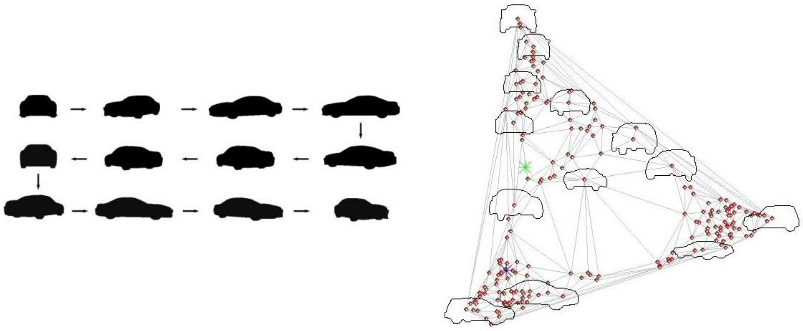

Figure 8: a) 12 shapes for one of the 17 cars used in the dataset. b) Reduced space of the car data set and its Delaunay triangulation.

define the speed function $F(\mathcal{K}, \mathcal{I}, v_{\mathcal{M}})$ as in [23] with a further dependency on the shape prior. This yields the following evolution equation

$$\frac{\partial \mathbb{D}_s(x,y)}{\partial \tau} = -(g(\nu + \mathcal{K}(x,y))|\nabla \mathbb{D}(x,y)| + \nabla \mathbb{D}(x,y) \cdot \upsilon_{\mathcal{M}}(x,y)). \tag{41}$$

(42)

Note that at each step of the evolution, we have to align the shape with the training samples using the principal moments before computing its embedding and deriving the shape prior term $\upsilon_{\mathcal{M}}$. The data term designed to attract the curve towards image edges [28] push or pulls the contour towards edges because of the stopping function $g(\nabla \mathcal{I}) = \frac{1}{1+||\nabla \mathcal{I}||^2}$. And ν is a constant baloon force as in [24]. In order to demonstrate the influence of our shape prior, we achieved segmentation of partly occluded cars which are not in the initial data set. We also chose images whose points of views are completely different. We initialized the contour with an ellipse around the car, with and without our shape prior, to segment and observe the evolution in both cases. The final results are presented in Fig. 9). Without the shape prior, the energy is obviously minimized on the image edges. However, when the shape prior is incorporated, the new energy overcomes local minima of the data term energy and finally computes a more accurate segmentation.

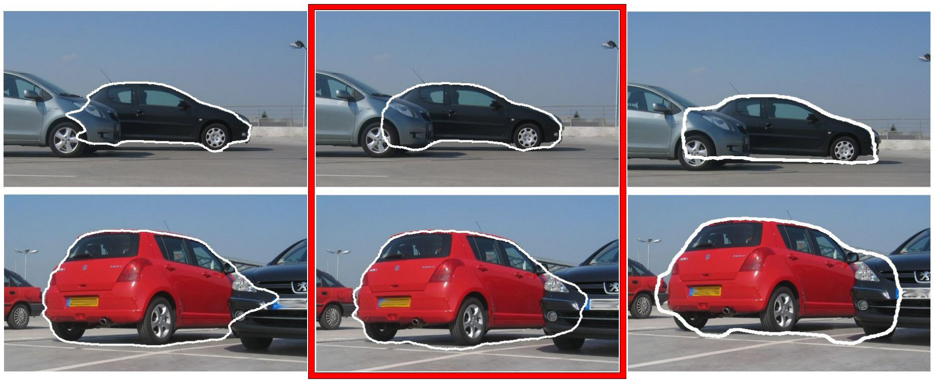

Figure 9: Segmentation of a Peugeot 206 (first row) and a Suzuki Swift (second row). **First column:** Segmentation with data term only. **Second column:** segmentation with our shape prior. The embedding of the final shape is denoted by a blue cross and a green cross respectively for the Peugeot 206 and the Suzuki Swift in Fig. 8 b) **Third column:** Segmentation with the nearest neighbor in the shape space as prior (such choice is not relevant compared to the nearest neighbors in the diffusion coordinates

5. Conclusion

In this paper, we focused on diffusion maps as a framework for shape modeling and processing. Our contribution is threefold. First, we use Nyström extensions [3] which provide a sound and efficient framework for extending embedding coordinates to the full shape or image space of possibly infinite dimension. Second, we formulated a variational framework based on distance minimization between the nearest neighbors in feature space. We applied our variational formulation to several applications such as the computation of the pre-image for denoising and image segmentation with a non-linear energy term. Results on real world data, such as 3D shapes and noisy 2D images, demonstrated the superiority of our approach over KPCA methods or linear energy terms.

In the continuation of this work several ideas may be exploited. In the perspective of working on complex shape spaces, our projection operator, defined from a manifold point-of-view, could be used in different tasks in interpolation and reconstruction of shapes and manifold denoising. Interestingly, our approach is able to deal with manifolds of complex topology. In the context of manifold denoising this property can be useful. So far, none of the pre-image problems were tested when the training data itself contains noise. We are currently looking into these possibilities.

References

[1] Arias, P., Randall, G., Sapiro, G., 2007. Connecting the out-of-sample and pre-image problems in kernel methods. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[2] Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15 (6), 1373–1396.

[3] Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Le Roux, N., Ouimet, M., 2004. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In: Thrun, S., Saul, L. K., Schölkopf, B. (Eds.), Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA.

[4] Carreira-Perpiñan, M., Lu, Z., 2007. The Laplacian Eigenmaps Latent Variable Model. JMLR W&P 2, 59–66.

[5] Charpiat, G., Faugeras, O., Keriven, R., 2005. Approximations of shape metrics and application to shape warping and empirical shape statistics. Foundations of Computational Mathematics 5 (1), 1–58.

[6] Chung, F. R. K., 1997. Spectral graph theory. Published for the Conference Board of the Mathematical Sciences, Washington, DC.

[7] Coifman, R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., Zucker, S., 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. PNAS 102 (21), 7426–7431.

[8] Coifman, R. R., Lafon, S., 2006. Diffusion maps. Applied and Computational Harmonic Analysis 21 (1), 5 – 30.

[9] Cremers, D., Kohlberger, T., Schnörr, C., 2002. Nonlinear shape statistics in mumford shah based segmentation. In: European Conference on Computer Vision. pp. 93–108.

[10] Dambreville, S., Rathi, Y., Tannenbeaum, A., 2006. Shape-based approach to robust image segmentation using kernel pca. In: CVPR '06: Proceedings of the International Conference on Computer Vision and Pattern Recognition. pp. 977–984.

[11] Davis, B. C., Fletcher, P. T., Bullitt, E., Joshi, S., 2007. Population shape regression from random design data. In: ICCV.

[12] Dupuis, P., Grenander, U., 1998. Variational problems on flows of diffeomorphisms for image matching.

[13] Etyngier, P., Segonne, F., Keriven, R., Oct 2007. Shape priors using manifold learning techniques. In: 11th IEEE International Conference on Computer Vision. Rio de Janeiro, Brazil.

[14] Ham, J., Lee, D. D., Mika, S., Schölkopf, B., 2003. A kernel view of the dimensionality reduction of manifolds. Tech. Rep. 110, Max-Planck-Institut für Biologische Kybernetik, Tübingen, Germany.

[15] Hein, M., Audibert, J. Y., 2005. Intrinsic dimensionality estimation of submanifolds in $\mathbb{R}^d$. In: ICML '05: Proceedings of the 22nd international conference on Machine learning. ACM, New York, NY, USA, pp. 289–296.

[16] Hein, M., yves Audibert, J., Luxburg, U. V., 2005. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. In: Proceedings of the 18th Conference on Learning Theory (COLT). Springer, pp. 470–485.

[17] Karcher, H., 1977. Riemannian center of mass and mollifier smoothing. Comm. Pure Appl. Math (30), 509–541.

[18] Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: Active Contour Models. Academic Publishers.

[19] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Jr., A. Y., 1994. Gradient flows and geometric active contour models. In: International Conference on Computer Vision.

[20] Kwok, J. T., Tsang, I. W., 2003. The pre-image problem in kernel methods. In: ICML. pp. 408–415.

[21] Lafon, S., Keller, Y., Coifman, R. R., 2006. Data fusion and multicue data matching by diffusion maps. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11), 1784–1797.

[22] Lafon, S., Lee, A. B., 2006. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (9), 1393–1403.

[23] Leventon, M. E., Grimson, E., Faugeras, O., 2000. Statistical shape influence in geodesic active contours. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 316–323.

[24] Lorigo, L. M., Faugeras, O., Grimson, W., Keriven, R., Kikinis, R., 1998. Segmentation of bone in clinical knee mri using texture-based geodesic active contours. In: MICCAI.

[25] Lu, Z., Carreira-Perpinan, M., Sminchisescu, C., 2008. People tracking with the laplacian eigenmaps latent variable model. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (Eds.), Advances in Neural Information Processing Systems 20. MIT Press, Cambridge, MA, pp. 1705–1712.

[26] Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M., Rätsch, G., 1999. Kernel PCA and de–noising in feature spaces. In: Kearns, M. S.,

Solla, S. A., Cohn, D. A. (Eds.), Advances in Neural Information Processing Systems 11. MIT Press.

[27] Mount, D. M., Arya, S., 1997. Ann: A library for approximate nearest neighbor searching.

[28] Osher, S., Fedkiw, R., 2001. Level set methods: an overview and some recent results. Journal of Computational Physics 169 (2), 463–502.

[29] Osher, S., Paragios, N., July 2003. Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer.

[30] Pennec, X., July 2006. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. Journal of Mathematical Imaging and Vision 25 (1), 127–154.

[31] Rousson, M., Paragios, N., 2002. Shape priors for level set representations. In: ECCV'02: Proceedings of the European Conference on Computer Vision. Vol. 2. pp. 78–92.

[32] Roweis, S., Saul, L. K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.

[33] Solem, J., 2006. Geodesic curves for analysis of continuous implicit shapes. In: International Conference on Pattern Recognition. Vol. 1. pp. 43–46.

[34] Tenenbaum, J. B., de Silva, V., Langford, J. C., December 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323.

[35] Thorstensen, N., Segonne, F., Keriven, R., 2009. Diffusion maps for shape and image denoising. In: SSVM' 09: Proceedings of the 2009 Conference on Scale Spaces and Variational Methods in Computer Vision.