Eurographics Workshop on 3D Object Retrieval (2015)
I. Pratikakis, M. Spagnuolo, T. Theoharis, L. Van Gool, and R. Veltkamp (Editors)

Accelerating the computation of canonical forms for 3D nonrigid objects using Multidimensional Scaling

Gil Shamai[1] Michael Zibulevsky[1] Ron Kimmel[1]

[1]Computer Science Department, Technion, Israel Institute of Technology, Haifa 32000

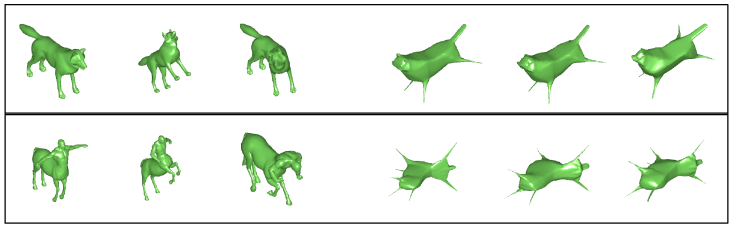


Figure 1: Shapes (left) from the TOSCA database [BBK08] and their corresponding canonical forms (right) obtained by the proposed Nyström Multidimensional Scaling.

Abstract

The analysis of 3D nonrigid objects usually involves the need to deal with a large number of degrees of freedom. When trying to match two such objects, one approach is to map the surfaces into a domain in which the matching process is simple to execute. Limiting the discussion to almost isometric mappings, which describe most natural deformations in nature, one could resort to Canonical forms. *Such forms translate the surface's intrinsic geometry into an extrinsic one in a Euclidean space, thus eliminating the effect of deformations at the expense of (hopefully) minor embedding errors. Multidimensional Scaling (MDS) is a dimensionality reduction technique that can be used to compute canonical forms of 3D-objects, by first evaluating the pairwise geodesic distances between surface points, and then embedding the distances in a lower dimensional Euclidean space. The native computational and space complexities involved in describing such inter-geodesic distances is quadratic in the number of surface points, a property that could be prohibiting in various scenarios. We present an acceleration framework for multidimensional scaling, by accurately approximating the pairwise distance maps. We show how the proposed Nyström Multidimensional Scaling (NMDS) framework can be used to compute canonical forms in quasi-linear time and linear space complexities in the number of data points. It allows us to efficiently deal with high resolution structures without giving up the embedding accuracy.*

1. Introduction

With the growth in the amount of digital information being stored and analyzed appears the need for simplification and dimensionality reduction. Methods, such as *principal component analysis* (PCA) [Wil01], *self-organizing map* (SOM) [Koh98], and *multidimensional scaling* (MDS) [BG05], are data reduction techniques that occupy the minds of researchers, who constantly try to reduce their computational and space complexities.

Multidimensional scaling (MDS) is one such dimensionality reduction technique, which attempts to map data into a low dimensional space. The data is being embedded while preserving, as much as possible, some affinity measure between each pair of data points. Such dimensionality reduction techniques were used, for example, to flatten numerical models of monkeys' cortical surfaces [SSW89] [DVEA*96], for texture mapping [ZKK02] [GKK02], and for image and video analysis [Sch01] [RT00] [AK06] [Ple03].

In [EK03], MDS was used to embed surface points of 3D

DOI: 10.2312/3dor.20151057

non-rigid shapes into $\mathbb{R}^3$, such that the shortest geodesic distance between each pair of points is close, as much as possible, to the Euclidean distance between corresponding embedded points. It is shown that the embedded set of points, known as a *Canonical form*, is invariant to isometric deformations of the 3D shapes. See Figure 1 for example. Canonical forms could thereby be used for non-rigid object recognition, classification, and shape matching. This task requires the computation of all pairwise geodesic distances, which can be time and space consuming and impractical when dealing with more than a few thousands of points. Efficient procedures such as the *fast marching method* [KS98], can compute the distance map between all pairs of points, in time complexity of $O(p^2 \log p)$, where p is the number of data points.

Here, we show how the computation of canonical forms can be accelerated, by projecting the geodesic distances onto a low dimensional subspace, which is learned from a small set of examples. This also allows to avoid storing the full pairwise distances matrix. Other attempts to accelerate MDS were made in [BBKY06], using multi-gird. When giving up the non-local structure of the data manifold and considering only relations between nearby points, the space complexity is $O(p)$, see for example the Local Linear Embedding (LLE [RS00]) and Hessian Local Linear Embedding (HLLE [DG03]). In [WSBA07], dense point to point correspondence between two surfaces was found by first performing MDS on a group of landmark points, then by computing a rigid correspondence between the canonical forms, and finally by projecting the rest of the points to the embedding space. MDS is performed using SMACOF algorithm, which is iterative. The idea of first embedding a group of landmarks and then projecting the rest of the points was already proposed in Landmark-Isomap [ST02], and is performed in a simple closed form. Dealing with classical multidimensional scaling in the spectral domain was suggested in Spectral MDS (SMDS) [AK13]. It allows substantial reduction in complexity using the fact that the full inter-geodesics distances matrix can be reliably approximated from just a small subset of its rows (and columns). In fact, the flat embedding itself can be considerably accelerated by operating in the data laplacian eigen-space.

The Nyström method [WS01] [Pla05] is an efficient technique that can be used to construct low rank approximations of symmetric matrices, using only a few columns chosen randomly from the matrix. [AW10] reviews the Nyström method and proposes theoretical guarantees for random sampling. In [YZZY12], Nyström method was used for approximating the affinity matrix and to perform MDS. Instead of a random sampling, an incremental sampling scheme was proposed for choosing the columns one by one, such that the variance of the affinity matrix is minimized. [CMIBR07] used Nyström method for graph drawing, and proposed a different sampling scheme based on the *farthest point strategy*, which we find to be more efficient and provided better approximation results. A regularization term is then used for the pseudo-inverse computation, which further improves the approximation. The graph shortest path distances are being measured using BFS, which assumes all edges are with equal length. [LJZ06] combined Kernel-PCA [SSM98] with Nyström method and showed how it can be efficiently used for mesh segmentation and for finding mesh correspondence.

Here, inspired by Spectral-MDS and Nyström, we develop a low rank matrix approximation using a learning approach. Then, with a small modification, we derive a variant of Nyström approximation with a low-rank regularization term. In practice, we still find the regularization used in [CMIBR07] to obtain slightly better results, though it is slower to compute. We use the *farthest point sampling strategy*, and the *fast marching method* for measuring the geodesic distances when embedding the surfaces, which is more adequate for surfaces with arbitrary connectivity, than BFS.

The structure of the paper is as follows. In Section 2 we review the *Classical multidimensional scaling* method for embedding a surface into a Euclidean space, using the pairwise geodesic distances. Next, in Section 3 we present the Nyström method. We develop an alternative learning approach for approximating a matrix, and show how it can be used to accelerate MDS. Finally, in Section 4 we support the proposed method with experimental results, that are followed by conclusions.

2. Review of Classical Scaling

Classical scaling is a dimensionality reduction procedure which attempts to map a given data into a low dimensional Euclidean space, while preserving, as much as possible, some affinity measure between each pair of data points. Let $\mathcal{V} = \{v_1, v_2, ..., v_p\}$ be a set of p vertices, and D a $p \times p$ affinity matrix such that D_{ij} is some affinity measure between v_i, v_j. Classical scaling aims at finding an embedding of the data in a low dimensional Euclidean space: Given k, find a set of points $\{x_i\}_{i=1}^p$ in $\mathbb{R}^k$, such that the Euclidean distance between x_i and x_j is close as possible to D_{ij}.

$$\|x_i - x_j\|_2^2 \approx D_{ij}^2, \quad \forall i, j. \tag{1}$$

This equation can be written in matrix formulation as

$$JXX^TJ \approx -\frac{1}{2}JEJ, \tag{2}$$

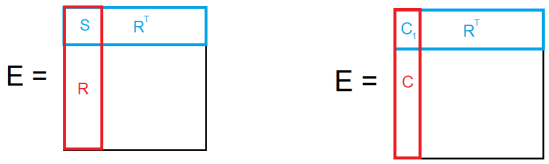
see [BG05] for more details. Here, $X_{p\times k}$ is a matrix holding the points $\{x_i\}_{i=1}^p$ in its rows, the matrix $E_{p\times p}$ is the element-wise square of D, i.e., $E_{ij} = D_{ij}^2$, and $J_{ij} = \delta_{ij} - \frac{1}{n}$ is a symmetric matrix, where as usual $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ii} = 1$ for all i. J is a centring matrix: $Z = JX$ is a translation of X to the origin, such that the mean of every column of Z is zero. Thus, computing Z instead of X, yields the same solution up to translation. Classical scaling formulates this problem through the minimization $\arg_Z \min \left\| ZZ^T + \frac{1}{2}JEJ \right\|_F$.

The solution for this problem is achieved by decomposing the symmetric matrix $-\frac{1}{2}JEJ$ into its eigenvalues and eigenvectors matrices UDV^T. Then, by considering the k largest eigenvalues and corresponding eigenvectors in the truncated matrices $\tilde{U}$ and $\tilde{D}$, the solution is given by $Z = \tilde{U}\tilde{D}^{\frac{1}{2}}$. The traditional classical scaling algorithm requires the computation of the full $E_{p\times p}$ matrix which is challenging to handle when dealing with more than several thousands of points. In the next sections we develop a matrix decomposition method which is efficient for matrices which are obtained through computation of geodesic distances. Later, we show how to use this method to efficiently solve the classical scaling problem and variants of it in quasi-linear time and linear space complexities.

3. Nyström Multidimensional Scaling

3.1. Nyström method for low-rank matrix approximation

In this section we derive a variant of the Nyström method for efficiently approximating the matrix E. Let E be a symmetric $p \times p$ matrix. The Nyström method is defined as follows: Randomly choose a set of n columns from E and let $R_{p\times n}$ contain these columns. Let $S_{n\times n}$ be the intersection of R and R^T in E. Figure 2a demonstrates the partition of E, where we assume without loss of generality that the chosen columns and corresponding rows are the first ones.



(a) (b)

Figure 2: Partition of E

The Nyström approximation of the matrix E is then given by $\hat{E} = RS^+R^T$, where S^+ is the pseudo-inverse of the matrix S. When the rank of E is n or less, and when S is a full rank matrix, it can be shown that this decomposition is the exact reconstruction of E. When this is not the case, but E can be approximated by a low rank matrix, we expect this decomposition to approximate E.

It is known that the pseudo-inverse of a matrix S can be obtained using its singular value decomposition. Denote by $S = W\Sigma U^T$ the *thin* singular value decomposition of S, i.e., the singular values equal to zero and corresponding singular vectors are removed. Then, the pseudo-inverse of S can be defined by $S^+ = U\Sigma^{-1}W^T$. When S is symmetric, the pseudo-inverse can be defined using the eigenvalue decomposition $S^+ = V\Lambda^{-1}V^T$, where $S = V\Lambda V^T$ is the *thin* eigenvalue decomposition of S. Nyström method is originally aimed to approximate strictly positive definite matrices. In our formulation, we justify the use of it for general symmetric matrices from a learning point of view.

3.1.1. Adding a regularization term

The smallest eigenvalues of S can cause instability in the computation of the pseudo-inverse, and a regularization is required. [MM84] proposed to use a regularized pseudo-inverse (RPI), which uses the expression

$$\frac{\sigma_i}{\sigma_i^2 + \alpha/\sigma_i^2}, \tag{3}$$

for the reciprocals in Σ^{-1}. Here, α is a parameter and σ_i are the singular values of S. In this manner, smaller singular values are increased by larger terms and hence their inverse is less sensitive to noise. α should be small enough to have a negligible effect on the large singular values, but large enough to affect the smaller ones. The authors in [CMIBR07] used this kind of regularization for the Nyström interpolation. They performed experiments and recommended choosing $\alpha = \sigma_1^3$, where σ_1 is the largest singular value. We propose a different regularization, which reduces the rank of S, hence also accelerates the computations. Denote by $\tilde{V}$ and $\tilde{\Lambda}$ the matrices which contain the $n_1 < n$ largest eigenvalues and corresponding eigenvectors of S. The alternative pseudo-inverse is then given by

$$S^+ = \tilde{V}\tilde{\Lambda}^{-1}\tilde{V}^T. \tag{4}$$

In this manner, the smallest $n - n_1$ eigenvalues are ignored and stability is obtained. Nevertheless, it is clear that n_1 should not be too small, so that we still keep enough eigenvectors that capture the structure of S. Figures 5a, 5b show the approximation error using both regularization terms, with respect to n_1 (CUR) and α (RPI).

3.1.2. A learning based approximation

In this section, we formulate the matrix approximation as a minimization problem and show how we derive the Nyström method with our proposed regularization discussed above, which is a low rank version of S^+. This formulation also explains the need for regularization from an over-fitting point of view.

Let E be a symmetric $p \times p$ matrix. Choose n columns of E and let R hold these columns. Define the matrix S as shown before in Figure 2a. Assuming the set of columns capture most of the information of E, a projection of E onto the subspace spanned by the columns could be used as a good low rank approximation for E, similar to projecting the data on the main principal components (PCA). The projection of E on the span of the columns of R is obtained by

$$\hat{E} = RR^+E. \tag{5}$$

In order to compute this projection, we would have to know

the entire matrix E. Since we only know a small subset of E (R and R^T), we suggest a different approach, in which we learn the projection using our available data. Since the projection is in the range of R, it can be defined through a coefficients matrix M as

$$\hat{E} = RM. \tag{6}$$

M is a $n \times p$ matrix, and we now attempt to find the entries of the matrix M. We propose the following learning approach: Choose $n_1 < n$ columns from R and let C hold these columns. Define C_1 as the intersection of C and R^T in E, as shown in Figure 2b, where without loss of generality we chose the first columns of R for C. C_1 is a $n \times n_1$ matrix. Now, solve the minimization problem

$$\arg_M \min \|C_1 M - R^T\|_F^2, \tag{7}$$

where now M is a $n_1 \times p$ matrix. This can be seen as using the known columns of E to learn M. This is a simple least squares problem. Its solution is given by

$$M = (C_1^T C_1)^{-1} C_1^T R^T = U R^T, \tag{8}$$

and our decomposition is formulated as

$$\hat{E} = CUR^T, \tag{9}$$

where $U = (C_1^T C_1)^{-1} C_1^T$. In this manner we have obtained what is known as a CUR decomposition of the matrix E. These kind of decompositions are widely used and studied, see [DKM06], [MD09] for example. Notice that we restrict the coefficient matrix $M_{n_1 \times p}$ to be smaller than originally defined in Equation (6). This helps us to avoid over-fitting of the model, as can be seen in Figure 5a (CUR), which measures the error $\|CM - E\|_F$ on the whole data, with respect to n_1. If we choose n_1 to be too small, we will not have enough columns in C to capture the subspace spanned by these columns of E. If we choose n_1 to be too large, we could over-fit the matrix R and the approximation error of the rest of the matrix will be large.

Let us introduce a small modification in the above procedure. Denote by $\tilde{V}$ and $\tilde{\Lambda}$ the matrices which contain the n_1 ($n_1 < n$) largest eigenvalues and corresponding eigenvectors of S. Now, instead of forming C from n_1 columns of R, we define $C = R\tilde{V}$. The columns of C are now linear combinations of the columns of R, instead of a subset of the R. The same minimization is then solved again, and in Figure 5a (Low-Rank) we see an improvement which is due to the better choice of C.

$$\hat{E} = CUR^T = C(C_1^T C_1)^{-1} C_1^T R^T \tag{10}$$

Notice that $C_1 = S\tilde{V} = V\Lambda V^T \tilde{V}$. Placing $C = R\tilde{V}$ and $C_1 = V\Lambda V^T \tilde{V}$ in the above equation, we obtain

$$\hat{E} = R\tilde{V}(\tilde{V}^T V\Lambda V^T V\Lambda V^T \tilde{V})^{-1} \tilde{V}^T V\Lambda V^T R^T, \tag{11}$$

which can be reduced to

$$\hat{E} = R\tilde{V}\tilde{\Lambda}\tilde{V}^T R^T = RTR^T. \tag{12}$$

This derivation can be thought of as a variant of the Nyström method where $T = \tilde{V}\tilde{\Lambda}\tilde{V}^T$ is used instead of S^+. T is a low rank matrix with rank n_1 and is nothing but the truncated eigenvalue decomposition of S^+ we proposed in Equation (4). In this manner we have derived the regularization term defined above, which can be seen as restricting the number of learned parameters to reduce over-fitting in a learning model.

3.2. Columns selection

The matrix decomposition developed in the previous section can be seen as a projection of E on the subspace spanned by the chosen columns. Hence, a good choice of columns would be one that capture the range of E to high accuracy with negligible failure probability. In this section, we suggest an iterative sampling strategy for choosing the columns, instead of choosing them randomly. The suggested strategy, known as the *farthest point sampling strategy*, is a method for selecting points that are far away from each other. the first point is selected at random. Then, at each iteration, the farthest point from the already selected ones is selected. It is known to be 2-optimal in sense of covering [HS85], and its complexity is $O(np \log p)$ when using *fast marching* for the distance computation on triangle meshes. The method is described in Procedure 1, for triangle meshes.

Procedure 1 *Farthest point sampling*

Input A triangle mesh with a set of p vertices $\mathcal{V} = \{v_1, v_2, ..., v_p\}$ and desired number of chosen vertices n.
Output a sampling $\mathcal{S} = \{r_1, ..., r_n\}$ and distances from the samples to the rest of the vertices $F_{p \times n}$

1: choose an initial vertex at random, $r_1 \leftarrow$ vertex index, $\mathcal{S} \leftarrow \mathcal{S} \cup \{r_1\}$
2: compute the geodesic distances from v_{r_1} to all vertices, $F_{(:,1)} \leftarrow dist(v_{r_1})$
3: **for** $i = 2$ to n **do**
4: find the farthest vertex from the already chosen ones, $r_i = \arg\max_{1 \le j \le |V|} \min_{1 \le k < i} F_{jk}$
5: update the set of selected samples, $\mathcal{S} \leftarrow \mathcal{S} \cup \{r_i\}$
6: compute the geodesic distances from v_{r_i} to all vertices, $F_{(:,i)} \leftarrow dist(v_{r_i})$
7: **end for**

Here, $dist(v)$ returns a vector of geodesic distances from vertex v to the rest of the vertices, using the *fast marching methods*. This procedure is being used as an efficient way of obtaining a few columns of the matrix E. Since the chosen samples are far, these columns are expected to capture most of the information of the matrix. While other existing methods need to store the whole matrix in memory or at least make a few passes on it to decide which columns are best to choose, here we do not need to know the entire matrix in advance. This is a strong advantage which could be exploited in problems related to pairwise geodesic computation.

3.3. Accelerating classical scaling

We present an efficient alternative for the classical scaling algorithm using the matrix approximation discussed above. Denote as before the approximation of E by: $\hat{E} = R\tilde{V}\tilde{\Lambda}^{-1}\tilde{V}^T R^T$, where $\tilde{V}$ is a $p \times n_1$ matrix and $\tilde{\Lambda}$ is a $n_1 \times n_1$ diagonal matrix. A straightforward solution would be similar to classical scaling. Namely, compute $\hat{A} = -\frac{1}{2}J\hat{E}J$, decompose it into UDU^T, and then form the truncated decomposition $\tilde{U}\tilde{D}\tilde{U}^T$, where $\tilde{D}$ and $\tilde{U}$ hold the k largest eigenvalues and k corresponding eigenvectors. The solution is then given by $Z = \tilde{U}\tilde{D}^{\frac{1}{2}}$. Since we would like to avoid computing and storing large matrices, the above straightforward procedure would fail to serve this purpose. For that goal, we propose the following efficient remedy. compute the QR factorization $QW = JR\tilde{V}$, where Q is an orthonormal $p \times n_1$ matrix, and W is an upper triangular $n_1 \times n_1$ matrix. Note that

$$\hat{A} = -\frac{1}{2}QW\tilde{\Lambda}^{-1}W^T Q^T. \tag{13}$$

Compute the k largest eigenvalues and eigenvectors of the symmetric matrix $-\frac{1}{2}W\tilde{\Lambda}^{-1}W^T$, using eigenvalue decomposition:

$$\tilde{V}_2\tilde{\Lambda}_2\tilde{V}_2^T = -\frac{1}{2}W\tilde{\Lambda}^{-1}W^T. \tag{14}$$

$\tilde{V}_2$ is a $n_1 \times k$ matrix containing the k eigenvectors that correspond to the k largest eigenvalues as its columns, and $\tilde{\Lambda}_2$ is a $k \times k$ matrix containing the k largest eigenvalues along its diagonal. We obtained the decomposition

$$\hat{A} = Q\tilde{V}_2\tilde{\Lambda}_2\tilde{V}_2^T Q^T. \tag{15}$$

It is clear that this is an eigenvalue decomposition of $\hat{A}$ since $Q\tilde{V}_2$ is orthonormal as a product of orhonormal matrices, and $\tilde{\Lambda}_2$ is diagonal. Therefore, we managed to obtain the eigenvalue decomposition without explicitly computing $\hat{A}$. Finally, the solution of the classical scaling problem is given by

$$Z = Q\tilde{V}_2\tilde{\Lambda}_2^{\frac{1}{2}}. \tag{16}$$

We sum up the final MDS acceleration in Procedure 2.

Procedure 2 *Nyström-MDS*

Input $G = \{E,V\}$ with $p = |V|$ vertices, parameters n, n_1.
Output A matrix Z which contains the coordinates of the embedding.

1: Choose n samples of the data using *farthest point sampling* and obtain the matrix F.
2: Compute R, the element-wise square of F, $R_{ij} = F_{ij}^2$.
3: Denote by S the intersection of R and R^T in E.
4: Compute $\tilde{V}$ and $\tilde{\Lambda}$, which contain the n_1 largest eigenvalues and corresponding eigenvectors of S, using eigenvalue decomposition
5: Compute $QW = JR\tilde{V}$ using QR factorization
6: Compute $\tilde{V}_2$ and $\tilde{\Lambda}_2$, which contain the k largest eigenvalues and corresponding eigenvectors of $-\frac{1}{2}W\tilde{\Lambda}^{-1}W^T$, using eigenvalue decomposition.
7: Return the coordinates matrix $Z = Q\tilde{V}_2\tilde{\Lambda}_2^{\frac{1}{2}}$

4. Results

Throughout this section, we compare between the Classical Scaling Method (MDS), Spectral MDS [AK13] (SMDS), and our proposed Nyström-MDS (NMDS) for embedding the affinity matrix of 3D shapes. We compare two methods for the geodesic distance computations, BFS used in [CMIBR07] and *fast marching on triangulated domains* [KS98] used here. We compare min-variance incremental sampling (Min-Var) used in [YZZY12] to the farthest point sampling strategy used here, for choosing the columns. We compare the regularization term used in [MM84](RPI) to the regularization term derived from our minimization problem. In addition, we compare our algorithm to Landmark-Isomap proposed by [ST02], and to the algorithm proposed by [LJZ06], which uses the Nyström method to efficiently perform Kernel PCA with gaussian radial basis function. When using SMDS, we use 300 eigenvectors.

Consider a non-rigid two dimensional surface in $\mathbb{R}^3$, described by a discrete connected triangle mesh with vertices. Let D be the affinity matrix which contains the pairwise geodesic distances between each two vertices on the surface. MDS can be used to embed D into a flat Euclidean space. This kind of embedding is known as a canonical form [EK03], and is invariant to isometric deformations of the surface. This idea can be exploited, for instance, for non-rigid objects recognition, eliminating the degree of freedom of isometric deformations. In the first example, we compute the canonical forms of two surfaces from the TOSCA database [BBK08], each defined by $p = 3400$ vertices. We apply the proposed procedure with $n_1 = 50$, choosing $n = 100$ columns using the farthest point sampling procedure, and using the *fast marching method* for the geodesic distances computation.

Figure 3 compares between canonical forms derived from MDS, NMDS, SMDS, NMDS using Min-Var instead of farthest point sampling, and NMDS using BFS instead of *fast marching*. As can be seen, the proposed NMDS provides similar canonical forms to the full MDS, with only $n = 10$ columns sampled from E. Since the mesh representing the shapes contains edges with varying lengths, BFS fails to measure the distances accurately. The corresponding embedding error is displayed under each canonical form.

Multidimensional Scaling can be used to visualize the inner structure of data, particularly when the data is a distance matrix. In the following example, we use MDS to visualize the classification of the canonical forms of 30 nonrigid shapes. We took 9 Cat shapes, 6 Centaur shapes and 15 David shapes and computed each of their canonical forms using NMDS and KPCA. We computed the Euclidean dis-

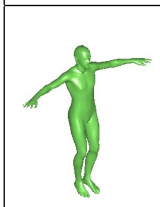
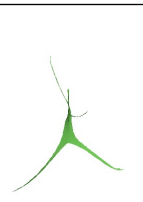
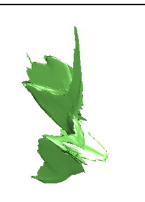
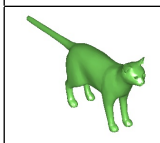
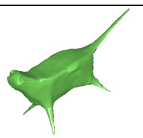
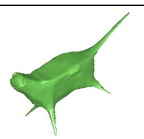
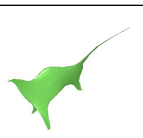
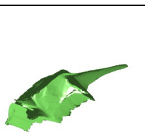
	MDS	NMDS	SMDS	Min-Var	BFS
					
Stress	394.41	398.1	640.6	4684.2	934.8
					
Stress	261.1	296	937.8	906.7	1184.4

Figure 3: Canonical forms of David and Cat, using only $n = 10$ ($n_1 = 5$) samples for the compared methods. Left to right: The original shape followed by canonical forms obtained by classical MDS, the proposed NMDS, SMDS, NMDS with Min-Var, and NMDS with BFS. The stress error $\frac{1}{p^2}\left\|ZZ^T + \frac{1}{2}JEJ\right\|_F$ of the embedding is displayed at the bottom of the corresponding form.

tances between each pair of the 30 canonical forms, and obtained a 30×30 pairwise Euclidean distance matrix. Finally, we embedded the distance matrix into $\mathbb{R}^2$ using MDS. Figure 4 shows the classification results. As can be seen, Canonical forms computed by NMDS allows us to classify the non-rigid objects. While Kernel-PCA is an efficient method that can be used to compute canonical forms, the gaussian filter used in [LJZ06] mainly considers local distances. Hence, the canonical forms are less distinctive, as can be seen in the classification.

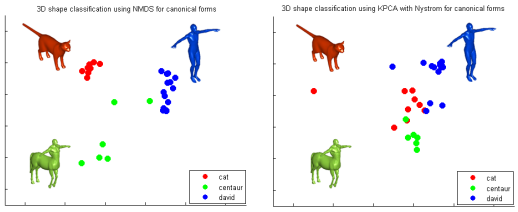
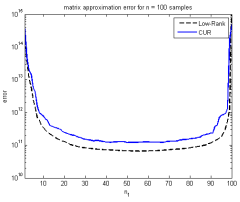
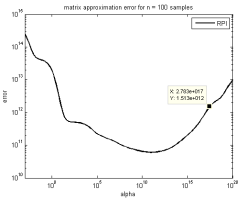


Figure 4: Classification of 30 shapes of Cat, David and Centaur, obtained by MDS on their Canonical forms, which were obtained by NMDS (left) and by KPCA with Nyström (right).

In figures 5a, 5b, 6 we measure the reconstruction error of the squared affinity matrix E, of the David shape, defined through the Frobenius norm $\frac{1}{p^2}\|\hat{E} - E\|_F^2$. In Figure 5a we plot the error of the CUR decomposition defined in Equation (9), and its modification defined in Equation (12) which we show to be equivalent to Nyström using (4) for the pseudo-inverse. In this plot, we fix the number of chosen columns n to 100 and change the value of n_1. In Figure 5b we plot the same error using the regularization defined in Equation (3) with respect to α, again fixing $n = 100$. A slightly better result is achieved for this regularization for the optimal α. Notice that the reconstruction error of Nyström reconstruction without a regularization term, as being used in [YZZY12], is obtained by choosing $n_1 = 100$ in Figure 5a or $\alpha = 0$ in Figure 5b. As can be seen, the regularization term significantly reduces the error.



(a) The error with respect to n_1, using the learning approach (CUR) and its modification which is equivalent to Nyström with the low rank regularization term (Low-Rank)



(b) The error with respect to α, using RPI regularization. We marked the choice of α according to the suggestion in [CMIBR07].

Figure 5: The affinity matrix reconstruction error

In Figure 6, we plot the above error as a function of the number of samples n, setting $n_1 = \left\lceil \frac{1}{2}n \right\rceil$ for NMDS. We again compare the results to the same procedure using BFS and Min-Var incremental sampling. Here, we refer the best rank n approximation of E as the ground truth, which is obtained by the solution of $\arg_M \min \|RM - E\|_F$, (the solution is given in Equation (5)).

In Figure 7, we measure the error of the final embedding,

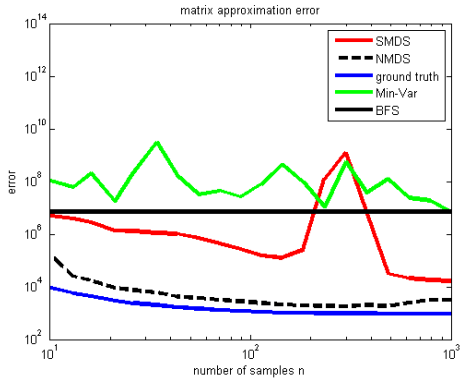


Figure 6: The affinity matrix reconstruction error with respect to the number of samples n, using different approaches.

computed by $stress(Z) - stress(Z^*)$, where

$$stress(Z) = \frac{1}{p^2}\|ZZ^T + \frac{1}{2}JEJ\|_F \qquad (17)$$

and Z^* is the embedding obtained by full MDS. We add the embedding error obtained by Landmark-Isomap (IS-ISOMAP). In this method, a group of landmarks is first selected and embedded using classical MDS. Then, the rest of the points are projected onto the subspace spanned by the embedded landmarks. This method is effective but limited, since the embedding subspace is determined only by the landmark points.

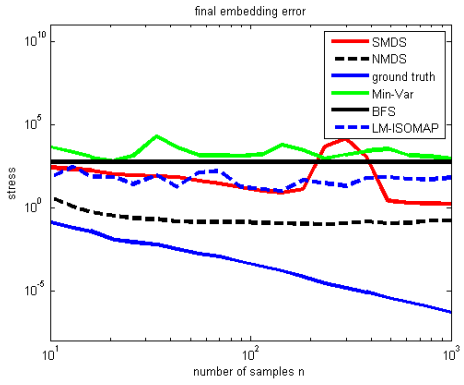


Figure 7: The final embedding error with respect to the number of samples n, after subtracting the optimal stress error obtained by Classic MDS, as defined in Equation (17).

In Figure 8, we measure the computation times of SMDS, NMDS with the low-rank regularization, no regularization term, and the RPI regularization term, on a shape with 3400 vertices. Here, we do not include the computation times needed for acquiring the geodesic distances. We plot the time in seconds, as a function of n, when choosing $n_1 = \left\lceil \frac{1}{2}n \right\rceil$ in NMDS. The difference computation times for the different regularization terms stems from Step 5 in Procedure 2.

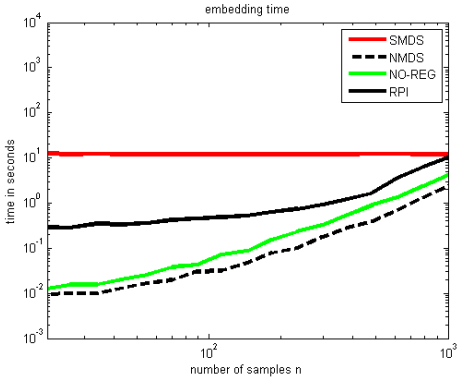


Figure 8: Computation times of NMDS with the low-rank regularization (NMDS), NMDS with no regularization term (NO-REG), and NMDS with the RPI regularization term(RPI), on a shape with 3400 vertices, not including the geodesic distances computation.

Finally, we run the whole procedure with $n = 10$ samples and compare the computation time to SMDS and MDS, on two shapes with 3400 and 28000 vertices. The algorithms were evaluated on an i5 Intel computer with 4GB RAM. The following table compares between the average time it took each of the methods to compute the result, including the computation of the geodesic distances. As for MDS, it is impossible to apply it to 28000 vertices due to memory limitations. With more memory the computation time would have taken more than a couple of hours.

# vertices	MDS	NMDS	SMDS
3400	25.6	0.24	12.5
28000	NaN	1.67	75.4

Figure 9: Computation times (in seconds) for MDS, NMDS, and SMDS on shapes with 3400 and 28000 vertices.

5. Conclusions

We formulate a minimization problem based on a learning approach for finding a low-rank approximation of the affinity matrix used in MDS. We show its connection to the Nyström method, and explain the need for a regularization term from an over-fitting point of view. We exploit the inner dependencies of the pairwise geodesic distances and use the *farthest point sampling strategy* to choose representative columns, without knowing the entire matrix. The chosen columns are far from each other, and hence are expected to capture most of the action of the matrix. We use the *fast marching method* for an accurate computation of the geodesic distances and show that it leads to better results than with using BFS.

Finally, we pack the whole algorithm in a simple and fast procedure for Multidimensional Scaling approximation, and show how it can be used to obtain the canonical forms of 3D shapes.

6. Acknowledgements

The research leading to these results has received funding from the European Research Council under European Union's Seventh Framework Programme, ERC Grant agreement no. 267414

References

[AK06] AHARON M., KIMMEL R.: Representation analysis and synthesis of lip images using dimensionality reduction. *International Journal of Computer Vision 67*, 3 (2006), 297–312. 1

[AK13] AFLALO Y., KIMMEL R.: Spectral multidimensional scaling. *Proceedings of the National Academy of Sciences 110*, 45 (2013), 18052–18057. 2, 5

[AW10] ARCOLANO N., WOLFE P. J.: Nyström approximation of wishart matrices. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (2010), IEEE, pp. 3606–3609. 2

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical geometry of non-rigid shapes*. Springer, 2008. 1, 5

[BBKY06] BRONSTEIN M. M., BRONSTEIN A. M., KIMMEL R., YAVNEH I.: Multigrid multidimensional scaling. *Numerical linear algebra with applications 13*, 2-3 (2006), 149–171. 2

[BG05] BORG I., GROENEN P. J.: *Modern multidimensional scaling: Theory and applications*. Springer, 2005. 1, 2

[CMIBR07] CIVRIL A., MAGDON-ISMAIL M., BOCEK-RIVELE E.: SSDE: Fast graph drawing using sampled spectral distance embedding. In *Graph Drawing* (2007), Springer, pp. 30–41. 2, 3, 5, 6

[DG03] DONOHO D. L., GRIMES C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences 100*, 10 (2003), 5591–5596. 2

[DKM06] DRINEAS P., KANNAN R., MAHONEY M. W.: Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing 36*, 1 (2006), 184–206. 4

[DVEA*96] DRURY H. A., VAN ESSEN D. C., ANDERSON C. H., LEE C. W., COOGAN T. A., LEWIS J. W.: Computerized mappings of the cerebral cortex: a multiresolution flattening method and a surface-based coordinate system. *Journal of cognitive neuroscience 8*, 1 (1996), 1–28. 1

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*, 10 (2003), 1285–1295. 1, 5

[GKK02] GROSSMANN R., KIRYATI N., KIMMEL R.: Computational surface flattening: a voxel-based approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 4 (2002), 433–441. 1

[HS85] HOCHBAUM D. S., SHMOYS D. B.: A best possible heuristic for the k-center problem. *Mathematics of operations research 10*, 2 (1985), 180–184. 4

[Koh98] KOHONEN T.: The self-organizing map. *Neurocomputing 21*, 1 (1998), 1–6. 1

[KS98] KIMMEL R., SETHIAN J. A.: Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences 95*, 15 (1998), 8431–8435. 2, 5

[LJZ06] LIU R., JAIN V., ZHANG H.: Sub-sampling for efficient spectral mesh processing. In *Advances in Computer Graphics*. Springer, 2006, pp. 172–184. 2, 5, 6

[MD09] MAHONEY M. W., DRINEAS P.: CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences 106*, 3 (2009), 697–702. 4

[MM84] MAEDA J., MURATA K.: Restoration of band-limited images by an iterative regularized pseudoinverse method. *JOSA A 1*, 1 (1984), 28–34. 3, 5

[Pla05] PLATT J. C.: Fastmap, metricmap, and landmark mds are all Nyström algorithms. In *Proc. 10th Int. Workshop on Artificial Intelligence and Statistics* (2005), pp. 261–268. 2

[Ple03] PLESS R.: Image spaces and video trajectories: Using isomap to explore video sequences. In *ICCV* (2003), vol. 3, pp. 1433–1440. 1

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323–2326. 2

[RT00] RUBNER Y., TOMASI C.: *Perceptual metrics for image database navigation*. Springer, 2000. 1

[Sch01] SCHWEITZER H.: Template matching approach to content based image indexing by low dimensional euclidean embedding. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 2, IEEE, pp. 566–571. 1

[SSM98] SCHÖLKOPF B., SMOLA A., MÜLLER K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation 10*, 5 (1998), 1299–1319. 2

[SSW89] SCHWARTZ E. L., SHAW A., WOLFSON E.: A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 11*, 9 (1989), 1005–1008. 1

[ST02] SILVA V. D., TENENBAUM J. B.: Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems* (2002), pp. 705–712. 2, 5

[Wil01] WILLIAMS C. K.: On a connection between kernel PCA and metric multidimensional scaling. In *Advances in Neural Information Processing Systems 13* (2001), MIT Press, pp. 675–681. 1

[WS01] WILLIAMS C., SEEGER M.: Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems* (2001), no. EPFL-CONF-161322, pp. 682–688. 2

[WSBA07] WUHRER S., SHU C., BOSE P., AZOUZ Z. B.: Posture invariant correspondence of incomplete triangular manifolds. *International Journal of Shape Modeling 13*, 02 (2007), 139–157. 2

[YZZY12] YU H., ZHAO X., ZHANG X., YANG Y.: ISOMAP using Nyström method with incremental sampling. *Advances in Information Sciences & Service Sciences 4*, 12 (2012). 2, 5, 6

[ZKK02] ZIGELMAN G., KIMMEL R., KIRYATI N.: Texture mapping using surface flattening via multidimensional scaling. *Visualization and Computer Graphics, IEEE Transactions on 8*, 2 (2002), 198–207. 1