This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

Spherical and Hyperbolic Embeddings of Data

Richard C. Wilson, *Senior Member, IEEE,* Edwin R. Hancock *Senior Member, IEEE,*, Elżbieta Pękalska, Robert P. W. Duin *Member, IEEE,*

***Abstract*—Many computer vision and pattern recognition problems may be posed as the analysis of a set of dissimilarities between objects. For many types of data, these dissimilarities are not Euclidean (i.e. they do not represent the distances between points in a Euclidean space), and therefore cannot be isometrically embedded in a Euclidean space. Examples include shape-dissimilarities, graph distances and mesh geodesic distances. In this paper, we provide a means of embedding such non-Euclidean data onto surfaces of constant curvature. We aim to embed the data on a space whose radius of curvature is determined by the dissimilarity data. The space can be either of positive curvature (spherical) or of negative curvature (hyperbolic). We give an efficient method for solving the spherical and hyperbolic embedding problems on symmetric dissimilarity data. Our approach gives the radius of curvature and a method for approximating the objects as points on a hyperspherical manifold without optimisation. For objects which do not reside exactly on the manifold, we develop a optimisation-based procedure for approximate embedding on a hyperspherical manifold. We use the exponential map between the manifold and its local tangent space to solve the optimisation problem locally in the Euclidean tangent space. This process is efficient enough to allow us to embed datasets of several thousand objects. We apply our method to a variety of data including time warping functions, shape similarities, graph similarity and gesture similarity data. In each case the embedding maintains the local structure of the data while placing the points in a metric space.**

***Index Terms*—embedding, non-Euclidean, spherical, hyperbolic**

I. Introduction

There are many problems in computer vision and pattern recognition which can be posed in terms of a set of measured *dissimilarities* between objects [15]. In other words, there are no intrinsic features or vectors associated with the objects at hand, but instead there is a set of dissimilarities between the objects. Some examples include shape-similarities, gesture interpretation, mesh geodesic distances and graph comparison, but there are many more. There are two challenges to the analysis of such data. First, since they are not characterized by pattern-vectors, the objects can not be clustered or classified using standard pattern recognition techniques. For example, pairwise rather than central clustering techniques must be used on such data. Alternatively, the objects can be embedded into a vector-space using techniques such as multidimensional scaling [6] or IsoMap [31]. Once embedded in such a space then the objects can be characterized by their embedding co-ordinate vectors, and analyzed in a conventional manner.

Most embedding methods produce an embedding that is Euclidean. However, dissimilarity data cannot always be embedded exactly into a Euclidean space. This is the case when the symmetric similarity matrix (the equivalent of a kernel matrix) contains negative eigenvalues. Examples of such dissimilarity data occur in a number of data sources furnished by applications in computer vision. For instance, shape-similarity measures and graph-similarity measures are rarely Euclidean. Previous work [22] has shown that there is potentially useful information in the non-Euclidean component of the dissimilarities. Such data can be embedded in a pseudo-Euclidean space, i.e. one where some dimensions are characterized by negative eigenvalues and the squared-distance between objects has both positive and negative components which sum together to give the total distance. A pseudo-Euclidean space is however non-metric, which makes it difficult to correctly compute geometric properties. An alternative, which we explore in this paper, is to embed the data on a curved manifold, which is metric but non-Euclidean. The use of a metric space is important because it provides the possibility of defining classifiers based on geometric concepts such as boundaries, regions and margins.

Riemannian manifolds offer an interesting alternative to Euclidean methods. A Riemannian manifold is curved, and the geodesic distances between points on the manifold are metric. However the distances can also be indefinite (in the sense that the similarity matrix is indefinite) and so can represent indefinite dissimilarities in a natural way. Our goal in this paper is to embed objects onto the constant curvature space and its associated spherical or hyperbolic geometry. This is a potentially very useful model since, although it has intrinsic curvature, geodesics are easily computed and the geometry is well understood.

In our formulation, the space can be either of positive curvature (i.e. a spherical surface) or of negative curvature (i.e. a hyperbolic surface). We show how to approximate a distribution of dissimilarity data by a suitable hypersphere. Our analysis commences by defining the embedding in terms of a co-ordinate matrix that minimises the Frobenius norm with a similarity matrix. We show how the curvature of the embedding hypersphere is related to the eigenvalues of this matrix. In the case of a spherical embedding, the radius of curvature is given by an optimisation problem involving the smallest eigenvalues of a similarity matrix, while in the case of a hyperbolic embedding it is dependent on the second-smallest eigenvalue. Under the embedding, the geodesic distances between points are metric but non-Euclidean. Once embedded, we can characterize the objects using a revised dissimilarity matrix based on geodesic distances on the hypersphere. We

R. C. Wilson and E. R. Hancock are with the Department of Computer Science, University of York, UK

E. Pękalska is with the School of Computer Science, University of Manchester, United Kingdom

R. P. W. Duin is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Netherlands

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). Edwin Hancock was also supported by a Royal Society Wolfson Research Merit Award.

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

apply our method to a variety of data including time warping functions, shape similarities, graph comparison and gesture interpretation data. In each case the embedding maintains the local structure of the data while placing the points in a metric space.

II. Related Work

Multidimensional scaling (MDS) has its roots in Psychometrics and has a long history. Initially, the goal was to analyze perceptual similarities in order to visualize the results of psychological experiments. In classical MDS, the embedding space is generally Euclidean and an embedding space often sought which is two or three dimensional for visualization purposes. However, it was soon realized that some types of data do not seem to lie naturally on a Euclidean manifold [29]; the perceptual similarities of color and musical notes are good examples. Rather, these seem to lie on curves or circles in the embedding space.

Over the last decade or so, there has been a resurgence in research into embedding techniques, fueled by the appearance of a family of spectral embedding methods, typified by ISOMAP [31], Laplacian Eigenmaps [2] and Locally Linear Embedding (LLE) [25]. These are non-linear embedding techniques, which are able to embed points from a non-linear high-dimensional manifold into a low-dimensional space while preserving the structure. The essence of these methods is to examine the structure of the points using local neighbourhoods, and to preserve this structure in the final embedding. While these methods successfully embed non-linear data-manifolds, the final embedding is Euclidean. They therefore cannot represent manifolds with non-zero *intrinsic* curvature (as opposed to extrinsically curved manifolds).

To overcome some of the limitations imposed by a Euclidean embedding space, a family of methods has been proposed which model the data-manifold as a set of piecewise connected manifolds. Local Tangent Space Alignment(LTSA) [36] and manifold charting [3] are examples of this approach. As an example, LTSA discovers a set of local tangent spaces and the neighbourhood relationships between them, via a set of global to local transformations. Such methods can approximate intrinsically curved manifolds, but can be computationally expensive. The piecewise model also makes computations such as the geodesic distance between arbitrary points more complex.

Motivated by the observation that many datasets seem to lie on arcs or circles, there have been a number of works which look at the problem of embedding dissimilarities onto curved manifolds, most typically circles or spheres. For example, Hubert et al [16] have investigated the use of unidimensional embeddings on circles. In particular, the problem of mapping distances onto the sphere S^2 has received particular attention since it has a number of applications such as the embedding of points on the surface of the Earth, and texture mapping spheroid objects [12]. Cox and Cox [7] were some of the first to look in detail at the problem of spherical embedding. They employ the Kruskal stress [17] of the point configuration and use spherical-polar coordinates to parameterise the point-positions. The stress can then be optimized with respect to the zenith and azimuth angles of the points. Similarly, Elad et al [12] use a stress measure which is then optimized with respect to the spherical polar coordinates of the points (Equation 1).

$$\epsilon = \sum_{i,j}^{n} w_{ij}(D_G(i,j) - D_G^E(i,j))^2 \tag{1}$$

These methods are effective and specifically designed for the two-dimensional sphere S^2. However, they do not easily extend to spheres of higher dimension.

These methods all follow a pattern which is typically of approaches to non-Euclidean MDS. The key idea is to define a measure of the quality of the embedding, called the stress, and then optimize the position of the points to minimize the stress. This is a very general approach which can be used to embed into all kinds of manifold. The optimization is an important step; here the stress majorization(SMACOF) algorithm has proved very popular [9], [10]. For more details of these approaches to MDS, readers are referred to [6]

The possibility of embedding onto higher dimensional spheres has been explored by Lindman and Caelli in the context of interpreting psychological data [19]. As with other methods, their method involves optimizing a stress which is an extension of the original MDS method of Torgerson [32]. Interestingly, Lindman and Caelli note that the mathematics of hyperbolic space is very similar to that of spherical space, and propose a method for embedding into hyperbolic space as well. This suggests that hyperbolic space may also be a viable alternative for representing dissimilarity-based data, although problems may arise from the different topologies - for example spherical space is closed, whereas hyperbolic space is not. Hyperbolic embeddings have also been explored by Shavitt and Tankel, who have used the hyperbolic embedding as a model of internet connectivity [28]. In other work, Robles-Kelly and Hancock [24] show how to preprocess the available dissimilarity data so that it conforms either to spherical or hyperbolic geometry. In practice the former corresponds to a scaling of the distance using a sine function, and the latter scaling the data using a hyperbolic sine function.

Non-Euclidean data has recently been receiving increasing attention. In particular, hyperspherical data has found application in many diverse areas of computer vision and pattern recognition. In the spectral analysis of materials, spectra are commonly length-normalized and therefore exist on a hypersphere. Similarly, it is common to use bag-of-words descriptors in document analysis along with the cosine distance, which is implicitly a spherical representation. In computer vision, a wide range of problems involving probability density functions, dynamic time-warping and non-rigid registration of curves can be cast in a form where the data is embedded on a hypersphere [30]. In [33], Veeraraghavan et al demonstrate the utility of modelling time-warping priors on a sphere for activity recognition. Spherical embedding therefore has a central role in many problems.

In this paper, we propose a number of novel extensions to address the problem of embedding into spherical and hyperbolic spaces. Firstly, we show how to find and appropriate radius of curvature for the manifold directly from the

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

data. We then develop a method of embedding into these spaces which, in contrast to other approaches, is not based on optimization. Finally, we develop an optimization scheme to refine the results which is specifically tailored to the problem of constant-curvature embeddings and easily extends to any number of dimensions in spherical or hyperbolic space.

III. Indefinite spaces

We begin with the assumption that we have measured a set of dissimilarities between all pairs of patterns in our dataset. This information is represented by the matrix $\mathbf{D}$, where D_{ij} is dissimilarity between the objects indexed by i and j. We can define an equivalent set of similarities by using the matrix of squared dissimilarities $\mathbf{D}'$, where $D'_{ij} = D^2_{ij}$. This is achieved by identifying the similarities as $-\frac{1}{2}\mathbf{D}'$ and centering the resulting matrix:

$$\mathbf{S} = -\frac{1}{2}(\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{D}'(\mathbf{I} - \frac{1}{n}\mathbf{J}) \tag{2}$$

Here $\mathbf{J}$ is the matrix of all-ones, and n is the number of objects. In Euclidean space, this procedure gives the inner-product or kernel matrix for the points.

If $\mathbf{S}$ is positive semi-definite, then the original dissimilarities are Euclidean and we can use the kernel embedding to find position-vectors $\mathbf{x}_i$ for the points in Euclidean space; Let the matrix $\mathbf{X}$ be the matrix of point-positions, such that the position-vector $\mathbf{x}_i$ of the i^{th} point corresponds to the i^{th} row of $\mathbf{X}$. Then we have

$$\mathbf{X} = \mathbf{U}_S\sqrt{\mathbf{\Lambda}_S}$$

where $\mathbf{U}_S$ and $\mathbf{\Lambda}_S$ are the eigenvector and eigenvalue matrices of $\mathbf{S}$, respectively. In this case, the relationship between the squared distance matrix and the kernel matrix is

$$D'_{ij} = S_{ii} + S_{jj} - 2S_{ij} \tag{3}$$

This is precisely the procedure used in classical MDS.

If $\mathbf{S}$ is indefinite, which is often the case, then the objects cannot exist in Euclidean space with the given dissimilarities represented as Euclidean distances. In this case $\mathbf{S}$ is not a kernel matrix. This does not necessarily mean the the dissimilarities are non-metric; metricity is a separate issue which we discuss below. One measure of the deviation from definiteness which has proved useful is the negative eigenfraction(NEF) [22] which measures the fractional weight of eigenvalues which are negative:

$$\text{NEF} = \frac{\sum_{\lambda_i<0}|\lambda_i|}{\sum_i|\lambda_i|} \tag{4}$$

If NEF=0, then the data is Euclidean.

We can assess the *non-metricity* of the data by measuring violations of metric properties. It is very rare to have an initial distance measure which gives negative distance, so we will assume than the dissimilarities are all positive. The two measures of interest are then the fraction of triples which violate the triangle inequality (TV) and the degree of asymmetry of the dissimilarities. The methods applied in this paper assume symmetry - some of the data we have studied shows mild asymmetry which is corrected before processing. We give figures in the experimental section for triangle violations and asymmetry.

One way to treat such non-Euclidean data is to correct it before embedding. An example of this is to disregard the negative eigenvalues present in $\mathbf{S}$. We then obtain

$$\mathbf{S}_+ = \mathbf{U}_S\mathbf{\Lambda}_+\mathbf{U}_S^T \tag{5}$$

where $\mathbf{\Lambda}_+$ is the eigenvalue matrix with negative eigenvalues set to zero. Now $\mathbf{S}_+$ is a kernel matrix and we can find the embedding in the standard way. We refer to this as the *kernel embedding* of $\mathbf{S}$ to highlight its derivation from the kernel matrix, but essentially this is identical to classical multidimensional scaling.

Another alternative is to embed the non-Euclidean dissimilarity data in a non-Riemannian pseudo-Euclidean space [21], [15]. This space uses the non-Euclidean inner product

$$< \mathbf{x}, \mathbf{y} >= \mathbf{x}^T\mathbf{M}\mathbf{y},\ \mathbf{M} = \begin{pmatrix} \mathbf{I}_p & 0 \\ 0 & -\mathbf{I}_q \end{pmatrix} \tag{6}$$

where $\mathbf{I}_a$ denotes the $a \times a$ identity matrix. The diagonal values of -1 in $\mathbf{M}$ indicate dimensions corresponding to the 'negative' part of the space. The space has a signature (p, q) with p positive dimensions and q negative dimensions. This inner-product induces a norm, or distance measure:

$$|\mathbf{x}|^2 =< \mathbf{x}, \mathbf{x} >= \mathbf{x}^T\mathbf{M}\mathbf{x} = \sum_{i_+} x_i^2 - \sum_{i_-} x_i^2 \tag{7}$$

where i_+ runs over the positive dimensions of the space, and i_- runs over the negative dimensions. We can then write the similarity as

$$\mathbf{S} = \mathbf{U}_S\mathbf{\Lambda}_{S||}^{\frac{1}{2}}\mathbf{M}\mathbf{\Lambda}_{S||}^{\frac{1}{2}}\mathbf{U}_S^T \tag{8}$$

and the pseudo-Euclidean embedding is

$$\mathbf{X} = \mathbf{U}_S\mathbf{\Lambda}_{S||}^{\frac{1}{2}} \tag{9}$$

where $\mathbf{\Lambda}_{S||}$ indicates that we have take the absolute value of the eigenvalues.

So the pseudo-Euclidean embedding reproduces precisely the original distance and similarity matrices. However, while the pseudo-Euclidean embedding reproduces the original distance matrix, it introduces a number of other problems. The embedding space is non-metric and the squared-distance between pairs of points in the space can be negative. Locality is not preserved in such a space, and geometric constructions such as lines are difficult to define in a consistent way. The space is more general than needed to represent the given dissimilarities (as it allows negative squared-distances) and the projection of new objects is ill defined. In order to overcome these problems, we would like to embed the points in a space with a metric distance measure which produces indefinite similarity matrices; this means that the space must be curved.

IV. Geometry of Constant-curvature manifolds

A. Spherical Geometry

Spherical geometry is the geometry on the surface of a hypersphere. The hypersphere can be straightforwardly embedded in Euclidean space; for example the embedding of a

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

sphere of radius r in three dimensions is well known:

$$x^2 + y^2 + z^2 = r^2$$
$$\mathbf{x} = (r \sin u \sin v, r \cos u \sin v, r \cos v)^T \quad (10)$$

The embedding of an $(n-1)$-dimensional sphere in n-dimensional space is a straightforward extension of this.

$$\sum_i x_i^2 = r^2 \quad (11)$$

This surface is curved and has a constant sectional curvature of $K = 1/r^2$ everywhere.

The geodesic distance between two points in a curved space is the length of the shortest curve lying in the space and joining the two points. For a spherical space, the geodesic is a great circle on the hypersphere. The distance is the length of the arc of the great circle which joins the two points. If the angle subtended by two points at the centre of the hypersphere is θ_{ij}, then the distance between them is

$$d_{ij} = r\theta_{ij} \quad (12)$$

With the coordinate origin at the centre of the hypersphere, we can represent a point by a position vector $\mathbf{x}_i$ of length r. Since the inner product is $< \mathbf{x}_i, \mathbf{x}_j >= r^2 \cos\theta_{ij}$ we can also write

$$d_{ij} = r \cos^{-1}(\langle \mathbf{x}_i, \mathbf{x}_j \rangle / r^2) \quad (13)$$

The hypersphere is metric but not Euclidean. It is therefore a good candidate for representing points which produce indefinite kernels.

B. Hyperbolic geometry

As we previously observed, the pseudo-Euclidean(pE) space has been used to embed points derived from indefinite kernels. The pE space is clearly non-Riemannian as points may have negative distances to each other. However, it is still possible to define a sub-space which is Riemannian. As an example, take the 3D pE space with a single negative dimension (z) and the 'sphere' defined by

$$x^2 + y^2 - z^2 = -r^2$$
$$\mathbf{x} = (r \sin u \sinh v, r \cos u \sinh v, r \cosh v)^T \quad (14)$$

This space is called *hyperbolic* and is Riemannian. Distances measured on the surface are metric, even though the embedding pE space is non-Riemannian.

We can extend this hyperbolic space to more dimensions in a straightforward way:

$$\sum_{i_+} x_i^2 - z^2 = -r^2 \quad (15)$$

If there is more than one negative dimension, the surface is no longer Riemannian. The hyperbolic space is therefore restricted to any number of positive dimensions but just one negative dimension. Finally, the sectional curvature of this space, as with the hypersphere, is constant everywhere. In this case, the curvature is negative and given by $K = -1/r^2$. For the hyperbolic space, the geodesic is the analogue of a great circle. While the notion of angle in Euclidean space is geometrically intuitive, it is less so in pE space. However, we can define a notion of angle from the inner product. The inner product is defined as

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{k_+} x_{ik}x_{jk} - z_i z_j \quad (16)$$
$$= -|\mathbf{x}_i||\mathbf{x}_j| \cosh\theta_{ij}. \quad (17)$$

In four dimensions this is the familiar Minkowski inner product with signature $(+,+,+,-)$. This inner product defines the notion of hyperbolic angle. From this angle, the distance between two points in the space is $d_{ij} = r\theta_{ij}$. With the coordinate origin at the centre of the hypersphere, we can represent a point by a position vector $\mathbf{x}_i$ of length r. Since the inner product is $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = -r^2 \cosh\theta_{ij}$ we can also write

$$d_{ij} = r \cosh^{-1}(-\langle \mathbf{x}_i, \mathbf{x}_j \rangle / r^2) \quad (18)$$

V. Embedding

A. Embedding in spherical space

Given a distance matrix $\mathbf{D}$, we wish to find the set of points on a hypersphere which produce the same distance matrix. Since the curvature of the space is unknown, we must additionally find the radius of the hypersphere. We have n objects of interest, and therefore we would normally look for an n-1 dimensional Euclidean space. Since we have freedom to set the curvature, we must instead seek a $(n-2)$-dimensional spherical space embedded in the $(n-1)$-dimensional Euclidean space.

We begin by constructing a space with the coordinate origin at the centre of the hypersphere. If the point position-vectors are given by $\mathbf{x}_i, i = 1 \ldots n$, then we have

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = r^2 \cos\theta_{ij} = r^2 \cos(d_{ij}/r) \quad (19)$$

Next, we construct the matrix of point position-vectors $\mathbf{X}$, with each position-vector as a row. Then we have

$$\mathbf{X}\mathbf{X}^T = \mathbf{Z} \quad (20)$$

where $Z_{ij} = r^2 \cos(d_{ij}/r)$. Since the embedding space has dimension $n-1$, $\mathbf{X}$ consists of n points which lie in a space of dimension $n-1$ and $\mathbf{Z}$ should then be an n by n matrix which is positive semi-definite with rank $n-1$. In other words, $\mathbf{Z}$ should have a single zero eigenvalue, with the rest positive [27]. We can use this observation to determine the radius of curvature. Given a radius r and a distance matrix $\mathbf{D}$, we can construct $\mathbf{Z}(r)$ and find the smallest eigenvalue λ_1. By minimising the magnitude of this eigenvalue, we can find the correct radius of the hypersphere.

$$r^* = \arg\min_r |\lambda_1 [\mathbf{Z}(r)]| \quad (21)$$

In practice we locate the optimal radius via multisection search. The search is lower-bounded by the fact that the largest distance on the sphere is πr and therefore $r \geq d_{\min}/\pi$, and upper-bounded by $r \leq 3d_{\min}$ as the data is essentially Euclidean for such large radius (this is discussed below in more detail in Section V-B1). The smallest eigenvalue can be determined efficiently using the power method without the expense of performing the full eigendecompositon. After

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

the radius is determined, the embedding positions may be determined using the full eigendecomposition:

$$\mathbf{Z}(r^*) = \mathbf{U}_Z \mathbf{\Lambda}_Z \mathbf{U}_Z^T \quad (22)$$

$$\mathbf{X} = \mathbf{U}_Z \mathbf{\Lambda}_Z^{\frac{1}{2}} \quad (23)$$

This procedure can also be used to locate a subspace embedding with dimension less than $n-1$. If we wish to find an embedding space of dimension m, then we can try to minimise the remaining $n-m$ eigenvalues by finding $r^* = \arg\min_r \sum_{i \leq n-m} |\lambda_i [\mathbf{Z}(r)]|$.

If the points truly lie on a hypersphere, then this procedure is sufficient to correctly locate them. However, in general this is not the case. The optimal smallest eigenvalue λ_1 will be less than zero, and there will be residual negative eigenvalues. The embedding is then onto a 'hypersphere' of radius r, but embedded in a pseudo-Euclidean space. In order to obtain points on the hypersphere, we must correct the recovered points. The magnitude of the residual negative eigenvalues can be used to gauge how well the data conforms to spherical geometry; a small residual indicates the data is close to spherical.

The traditional method in kernel embedding is to discard the negative eigenvalues. Unfortunately, this will not suffice since this will change the length of the vectors and therefore the points will no longer lie on the hypersphere (constraint 11 will be violated). Although we can subsequently project the points back onto the hypersphere, in many cases this procedure proves unsatisfactory. In the next section we present an analysis of this problem and propose a solution.

B. Approximation of the points in spherical space

For a general set of dissimilarities, the points do not lie on a hypersphere, and their positions require correction to place them on the manifold. The conventional correction for a kernel embedding is to drop the negative eigenvalues with the corresponding dimensions. We show in the next section that this process is only justified for the spherical embedding in the case of a large radius of curvature. For more highly curved spaces, we propose a different approximation.

1) Limits of large radius: When the radius of curvature is large, clearly the manifold is nearly flat, and we might hope to recover the standard kernel embedding of the data. In fact we can write

$$\mathbf{Z} = r^2 \cos(\mathbf{D}/r) \simeq r^2 \mathbf{J} - \frac{1}{2}\mathbf{D}' \;\; (r >> 1) \quad (24)$$

As before, $\mathbf{J}$ is the matrix of all-ones. The squared distance matrix $\mathbf{D}'$ is related to the kernel matrix by

$$\mathbf{D}' = 2\mathbf{K}' - 2\mathbf{K} \quad (25)$$

where $K'_{ij} = (K_{ii} + K_{jj})/2$ is constructed from the diagonal elements of the kernel (Eqn 3), giving

$$\mathbf{Z} \simeq r^2 \mathbf{J} + \mathbf{K} - \mathbf{K}' \quad (26)$$

Since $\mathbf{K}$ is small compared to $r^2\mathbf{J}$ we can use degenerate eigenperturbation theory to show that $\mathbf{Z}$ and $\mathbf{K}$ have the same eigenvectors and eigenvalues. The exception is the leading eigenvalue of $\mathbf{Z}$, which is $\lambda_0 = nr^2 - \mathrm{Tr}(\mathbf{K})$, but the corresponding eigenvalue is zero for $\mathbf{K}$. As a result, we recover the kernel embedding for large r. This motivates us to use the standard approach of neglecting negative eigenvalues when embeddings onto the hypersphere with large radius($r >> 1$). As the resulting points no longer lie on the hypersphere, the final step is to renormalise the lengths of the position vectors to r.

2) Small radius approximation: We pose the problem as follows: The task is to find a point-position matrix $\mathbf{X}$ on the spherical manifold which minimises the Frobenius distance to the Euclidean-equivalent matrix $\mathbf{Z}$. Given the manifold radius r, determined by the method in the previous section, we begin with the normalised matrix $\hat{\mathbf{Z}} = \mathbf{Z}/r^2$. The problem is then

$$\min_{\mathbf{X}} \quad |\mathbf{X}\mathbf{X}^T - \hat{\mathbf{Z}}|$$
$$\mathbf{x}_i^T \mathbf{x}_i = 1 \quad (27)$$

This can be simplified in the usual way by observing that the Frobenius norm is invariant under an orthogonal similarity transform. Given $\hat{\mathbf{Z}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, we apply the matrix $\mathbf{U}$ as an orthogonal similarity transform to obtain the equivalent minimisation problem

$$\min_{\mathbf{X}} |\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U} - \mathbf{\Lambda}| \quad (28)$$

which has then solution $\mathbf{X} = \mathbf{U}\mathbf{B}$ where $\mathbf{B}$ is some diagonal matrix. The minimisation problem then becomes.

$$\min_{\mathbf{B}} |\mathbf{B}^2 - \mathbf{\Lambda}| \quad (29)$$

Of course, $\mathbf{B}^2 = \mathbf{\Lambda}$ is an exact solution if all the eigenvalues are non-negative, and this is the case if the points lie precisely on a hypersphere. In the general case, there will be negative eigenvalues and we must find a minimum of the constrained optimisation problem. In the constrained setting, we are no longer guaranteed that $\mathbf{B}$ should be a diagonal matrix. Nevertheless, here we make the simplifying approximation that we can find a diagonal matrix which gives a suitable approximate solution. This will be true if the points lie close to a hypersphere.

Let $\mathbf{b}$ be the vector of squared diagonal elements of $\mathbf{B}$, i.e. $b_i = B_{ii}^2$, and $\boldsymbol{\lambda}$ be the vector of eigenvalues. Finally U_s is the matrix of squared elements of $\mathbf{U}$, $U_{sij} = U_{ij}^2$. Then we can write the constrained minimisation problem as

$$\min_{\mathbf{b}} \quad (\mathbf{b} - \boldsymbol{\lambda})^T(\mathbf{b} - \boldsymbol{\lambda}) \quad (30)$$

$$b_i > 0 \; \forall i \quad (31)$$

$$\mathbf{U}_s \mathbf{b} = \mathbf{1} \quad (32)$$

While this is a quadratic problem, which can be solved by quadratic programming, the solution actually has a simple form which can be found by noting that the matrix $\mathbf{U}_s$ should have rank $n-1$ and hence one singular value equal to zero. To proceed we make the following observations: The vector of eigenvalues $\boldsymbol{\lambda}$ is an unconstrained minimiser of this problem, i.e. $\mathbf{b} = \boldsymbol{\lambda}$ minimises the Frobenius norm (and satisfies constraint (32)), but not constraint (31). Secondly, $\mathbf{b} = \mathbf{1}$ satisfies both constraints since $\sum_i U_{ij}^2 = 1$ (as $\mathbf{U}$ is orthogonal). These observations, together with the fact that

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

$\mathbf{U}_s$ is rank $n-1$, means that the general solution to the second constraint is

$$\mathbf{b} = \mathbf{1} + \alpha(\boldsymbol{\lambda} - \mathbf{1}) \tag{33}$$

It only remains then to find the value of α which satisfies constraint (31) and minimises the criterion. Since the criterion is quadratic, the solution is simply given by the largest value of α for which constraint (31) is satisfied. Given the optimal value α^* we can find $\mathbf{b}^*$ and

$$\mathbf{X}^* = \mathbf{U}\mathbf{B}^* \tag{34}$$

C. Embedding in Hyperbolic space

In hyperbolic space, we have

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = -r^2 \cosh \theta_{ij} = -r^2 \cosh(d_{ij}/r) \tag{35}$$

with the inner product defined by Eqn 6. Constructing $\mathbf{Z}$ as before, we get

$$\mathbf{X}\mathbf{M}\mathbf{X}^T = \mathbf{Z} \tag{36}$$

Again we have an embedding space of dimension $n-1$, but $\mathbf{Z}$ is no longer positive semi-definite. In fact, $\mathbf{Z}$ should have precisely one negative eigenvalue (since the hyperbolic space has just one negative dimension) and again a single zero eigenvalue. We must now minimise the magnitude of the second smallest eigenvalue to find the radius:

$$r^* = \arg\min_r |\lambda_2 [\mathbf{Z}(r)]| \tag{37}$$

The embedded point positions are now given by

$$\mathbf{X} = \mathbf{U}_Z \boldsymbol{\Lambda}_{Z||}^{\frac{1}{2}} \tag{38}$$

As with the spherical embedding, in general the points do not lie on the embedding manifold and there will be residual negative eigenvalues, beyond the single allowed negative eigenvalue.

D. Approximation of the points in hyperbolic space

A similar procedure to that used for the spherical space may also applied for hyperbolic space, but the optimisation problem is modified by the indefinite inner product. As with the spherical embedding, we may drop residual negative eigenvalues for large r. For small radius, the equivalent analysis is as follows. The matrix $\mathbf{M}$ is defined by Eqn. (6), with $q = 1$.

$$\begin{aligned} \min_{\mathbf{X}} \quad & |\mathbf{X}\mathbf{M}\mathbf{X}^T - \hat{\mathbf{Z}}| \\ \mathbf{x}_i^T \mathbf{M} \mathbf{x}_i \quad &= \quad -1 \end{aligned} \tag{39}$$

As before, we apply the orthogonal similarity transform given by $\mathbf{U}$, where $\hat{\mathbf{Z}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ to obtain the equivalent minimisation problem

$$\min_{\mathbf{X}} |\mathbf{U}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{U} - \boldsymbol{\Lambda}| \tag{40}$$

which has a solution $\mathbf{X} = \mathbf{U}\mathbf{B}$ where $\mathbf{D}$ is some diagonal matrix, giving the minimisation

$$\min_{\mathbf{B}} |\mathbf{B}\mathbf{M}\mathbf{B} - \boldsymbol{\Lambda}| \tag{41}$$

Now we have a vector of diagonal elements given by $b_i = (\mathbf{B}\mathbf{M}\mathbf{B})_{ii}$. Exactly one of the b_i's must be negative (the one corresponding to the most negative element of $\boldsymbol{\lambda}$). Let b_n be the component of $\mathbf{b}$ corresponding to the negative dimension. Finally, we then obtain a new constrained quadratic minimisation problem of

$$\min_{\mathbf{b}} \quad (\mathbf{b} - \boldsymbol{\lambda})^T(\mathbf{b} - \boldsymbol{\lambda}) \tag{42}$$

$$\begin{aligned} b_n \quad &< \quad 0 \\ b_i \quad &> \quad 0, \, \forall i \neq n \end{aligned} \tag{43}$$

$$\mathbf{U}_s\mathbf{b} \quad = \quad -\mathbf{1} \tag{44}$$

The value of $\mathbf{b} = \boldsymbol{\lambda}$ is global minimiser which satisfies constraint (44) and a second solution of the constraint is given by $\mathbf{b} = -\mathbf{1}$. We must therefore find the optimal value for α in the solution

$$\mathbf{b} = -\mathbf{1} + \alpha(\boldsymbol{\lambda} + \mathbf{1}) \tag{45}$$

The solution is more complicated than in the spherical case, due to the constraint $b_n < 0$. This means that it is possible that there is no solution (a case which is easily detected). If a solution exists, the optimal point will lie on one of the two boundaries of the feasible region. Given the optimal solution of α^*, we get $\mathbf{b}^*$ and $\mathbf{X}^* = \mathbf{U}\mathbf{B}^*$.

If there is not a feasible solution, this means that we cannot find a set of eigenvalues for the inner-product matrix $\mathbf{Z}^*$ which both satisfy the conditions that only one is negative and that have unit length. We must abandon one of these properties - in this case we return to our standard procedure of neglecting negative eigenvalues.

VI. Optimisation

The methods above provide the correct embeddings when the points lie exactly on the surface of a constant-curvature manifold, and a good approximation for points nearly on the manifold. Although the embeddings become unsatisfactory for larger approximations, they still provide a good initialisation for optimisation-based approaches. In this section, we develop an optimisation procedure based on the properties of the manifold. This method involves the greedy optimisation of a distance error function (essentially a stress-like measure). We apply the exponential map to transfer the optimisation to the tangent space where the updates are much more straightforward. This allows the method to extend elegantly to any number of dimensions. Under this map, the tangent space and manifold are locally isomorphic and so the computed gradients are the same and local minima of the error on the manifold are also local minima in the tangent space.

A. The Exponential Map

Non-Euclidean geometry can involve demanding calculations, and many problems are intractable on general Riemannian manifolds. However, by choosing a simple non-Euclidean manifold such as the hypersphere, we can hope to simplify problems such as embedding. To do so, we require one important tool of Riemannian geometry, which is the exponential map. The exponential map has previous found use in the statistical analysis of data on manifolds, for example in Principal Geodesic Analysis [13] and in the analysis of diffusion tensor data [23].

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

The exponential map is a map from points on the manifold to points on a tangent space of the manifold. The map has an origin, which defines the point at which we construct the tangent space of the manifold. The map has an important property which simplifies geometric computations; the geodesic distance between the origin of the map and a point on the manifold is the same as the Euclidean distance between the projections of the two points on the tangent space. As the tangent space is a Euclidean space, we can compute various geometric and statistical quantities in the tangent space in the standard way. Formally, the definition of these properties as follows: Let T_M be the tangent space at some point M on the manifold, P be a point on the manifold and X be a point on the tangent space. We have

$$\begin{aligned} X &= \mathrm{Log}_M P & (46)\\ P &= \mathrm{Exp}_M X & (47)\\ d_g(P,M) &= d_e(X,M) & (48)\end{aligned}$$

where $d_g(.,.)$ denotes the geodesic distance between the points on the manifold, and $d_e(.,.)$ is the Euclidean distance between the points on the tangent space.

The Log and Exp notation defines a log-map from the manifold to the tangent space and an exp-map from the tangent space to the manifold. This is a formal notation and does not imply the familiar $\log$ and $\exp$ functions. Although they do coincide for some types of data, they are not the same for the spherical space. The origin of the map M and is mapped onto the origin of the tangent space.

For the spherical manifold, the exponential map is as follows. We represent a point P on our manifold as a position vector $\mathbf{p}$ with fixed length $|\mathbf{p}| = r$ (the origin is at the centre of the hypersphere). Similarly, the point M (corresponding to the the origin of the map) is represented by the vector $\mathbf{m}$. The maps are then

$$\begin{aligned} \mathbf{x} &= \frac{\theta}{\sin\theta}(\mathbf{p} - \mathbf{m}\cos\theta) & (49)\\ \mathbf{p} &= \mathbf{m}\cos\theta + \frac{\sin\theta}{\theta}\mathbf{x} & (50)\\ d_g(P,M) &= r\theta = |\mathbf{x}| = d_e(X,M) & (51)\end{aligned}$$

where $\theta = \cos^{-1}\left(\langle \mathbf{p}, \mathbf{m}\rangle / r^2\right)$. The vector $\mathbf{x}$ is the image of P in the tangent space, and the image of M is at the origin of the tangent space.

For the hyperbolic manifold, the exponential map simply becomes

$$\begin{aligned} \mathbf{x} &= \frac{\theta}{\sinh\theta}(\mathbf{p} - \mathbf{m}\cosh\theta) & (52)\\ \mathbf{p} &= \mathbf{m}\cosh\theta + \frac{\sinh\theta}{\theta}\mathbf{x} & (53)\end{aligned}$$

where $\theta = \cosh^{-1}\left(\langle \mathbf{p}, \mathbf{m}\rangle / r^2\right)$. The required lengths and inner-products are calculated in the pseudo-Euclidean space (section IV-B).

B. Spherical Optimisation

Given a dissimilarity matrix $\mathbf{D}$, we want to find the embedding of a set of points on the surface of a hypersphere of radius r, such that the geodesic distances are as similar as possible to $\mathbf{D}$. Unfortunately, this is a difficult problem and requires an approximate optimisation-based approach. We simplify the problem by considering an incremental approach using just the distances to a single point at a time. Let the point of interest be $\mathbf{p}_i$; we then want to find a new position for this point on the hypersphere such that the geodesic distance to point j is d^*_{ij} where $*$ denotes that this is the target distance. We formulate the estimation of position as a least-squares problem which minimises the square error

$$E = \sum_{j\neq i}(d^2_{ij} - d^{*2}_{ij})^2 \quad (54)$$

where d_{ij} is the actual distance between the points. Direct optimisation on the sphere is complicated by the need to restrict points to the manifold. However, as we are considering a single point at a time, we can construct a linear embedding onto tangent space using the log-map and then optimise the positions in the Euclidean tangent space. If the current point-positions on the hypersphere are $\mathbf{p}_j, \forall j$, we can use the log-map to obtain point-positions $\mathbf{x}_j$ for each object j in the tangent space as follows:

$$\mathbf{x}_j = \frac{\theta_{ij}}{\sin\theta_{ij}}(\mathbf{p}_j - \mathbf{p}_i\cos\theta_{ij}) \quad (55)$$

with $\mathbf{x}_i = 0$.

We have found standard optimisation schemes to be infeasible on larger datasets, so here we propose a gradient descent scheme with optimal step-size determined by line-search. In this iterative scheme, we update the position of the point $\mathbf{x}_i$ in the tangent space so as to obtain a better fit to the given distances. At iteration k, the point is at position $\mathbf{x}_i^{(k)}$. Initially, the point is at the origin, so $\mathbf{x}_i^{(0)} = 0$. Since the points lie in tangent space, which is Euclidean, we then have

$$d^2_{ij} = (\mathbf{x}_j - \mathbf{x}_i)^T(\mathbf{x}_j - \mathbf{x}_i) \quad (56)$$

The gradient of the square-error function (Eqn. 54) is

$$\nabla E = 4\sum_{j\neq i}(d^2_{ij} - d^{*2}_{ij})(\mathbf{x}_i - \mathbf{x}_j) \quad (57)$$

and our iterative update procedure is

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \eta\nabla E \quad (58)$$

Finally, we can determine the optimal step size as follows: let $\Delta_j = d^2_{ij} - d^{*2}_{ij}$ and $\alpha_j = \nabla E^T(\mathbf{x}_i - \mathbf{x}_j)$, then the optimal step size is the smallest root of the cubic

$$\begin{aligned} &n|\nabla E|^4\eta^3 + 3|\nabla E|^2(\sum_j \alpha_j)\eta^2 \\ &+(2\textstyle\sum_j \alpha_j^2 + |\nabla E|^2\sum_j \Delta_j)\eta + \sum_j \alpha_j\Delta_j \end{aligned} \quad (59)$$

After finding a new point position $\mathbf{x}_i$, we apply the exp-map to locate the new point position on the spherical manifold.

$$\mathbf{p}'_i = \mathbf{p}_i\cos\theta + \frac{\sin\theta}{\theta}\mathbf{x}_i \quad (60)$$

The optimisation proceeds on a point-by-point basis until a stationary point of the squared-error E is reached.

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

C. Hyperbolic Optimisation

Optimisation on the hyperbolic manifold proceeds in a very similar way. However, in the hyperbolic case, we need to use the hyperbolic exponential map

$$\mathbf{x}_j = \frac{\theta_{ij}}{\sinh\theta_{ij}}(\mathbf{p}_j - \mathbf{p}_i\cosh\theta_{ij}) \tag{61}$$

with $\mathbf{x}_i = 0$. and bear in mind the inner product is modified by the pseudo-Euclidean embedding space. As a result, the squared distance is

$$d_{ij}^2 = (\mathbf{x}_j - \mathbf{x}_i)^T\mathbf{M}(\mathbf{x}_j - \mathbf{x}_i) \tag{62}$$

The gradient of the square-error function (Eqn. 54) is therefore

$$\nabla E = 4\sum_{j\neq i}(d_{ij}^2 - d_{ij}^{*2})\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \tag{63}$$

which gives $\alpha_j = \nabla E^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$. Additionally, the squared length of the gradient is $|\nabla E|^2 = \nabla E^T\mathbf{M}\nabla E$. With these ingredients, Equation (59) can be used without change to determine the optimal step size.

VII. Experiments

We investigate the efficacy of constant curvature embeddings using a variety of datasets including synthetic and real-world distances. Since scaling distances by a constant factor does not alter the geometry of the points, we first rescale the distance matrix so that the mean distance between points is 1. By doing this, we ensure that radii and distance errors are directly comparable between different datasets.

Our baseline comparison is with the kernel embedding (or classical MDS). For Euclidean distances, where the similarity matrix is positive semidefinite, this embedding is exact. For non-Euclidean distances, this is given by kernalising the similarity matrix (by eliminating negative eigenvalues):

$$\begin{aligned}\mathbf{S} &= \mathbf{U}_S\mathbf{\Lambda}\mathbf{U}_S^T \\ \mathbf{X}_K &= \mathbf{U}_S\mathbf{\Lambda}_+^{1/2}\end{aligned} \tag{64}$$

For experiments involving the surface of a sphere (S^2) we also compare our results to the spherical embedding method of Elad et al [12], which typifies the stress-minimising approach.

We use two measures of distortion for the embedded points. The first is the RMS difference between the distance matrix and the distances between the embedded points (in the embedding space). This measures the overall distortion introduced by the embedding and is a standard measurement. We have found in practice that although the RMS error measures embedding distortion, it does not reveal everything about the quality of the embedding for certain tasks such as classification and clustering. This is because there could be a local distortion which alters the local position of points close to each other, but which is small when measured over the whole space. This is essentially the motivation behind Locally Linear Embedding (LLE) [25] which embeds each local neighborhood in a linear space.

Since local configuration is important in some applications we have also used a measure of the change in neighbourhood order. This is achieved first choosing a central point and then ordering the points by distance away from the center and measuring the distortion in the ranked list using Spearman's rank correlation coefficient ρ. The structural error is $1-\rho$. This measure is then averaged over the choice of all points as the central point to obtain a structural error measure. This is zero if there is no changing is the distance-ranking of points, and one if the order is completely reversed.

A. Texture mapping

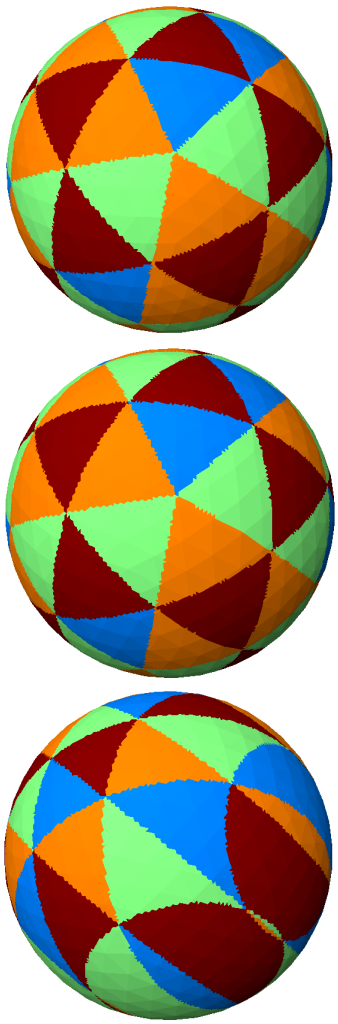

Fig. 1. Texture mapping of the sphere. Top: using our method, middle: the method of Elad et al, and bottom: kernel embedding. Both spherical methods produce virtually perfect embeddings, whereas the expected distortion is evident in the kernel embedding.

As an initial evaluation and comparison to the literature, we begin with a set of texture mapping experiments, similar to those conducted by Elad et al [12]. which involve a triangulated mesh describing the 3D surface of an object. Such mappings have been used by Bronstein et al [4] for representing face textures in a way that is more robust to expression changes. We then compute the geodesic distances across the mesh [20]. These distances between the vertices form the starting point for our algorithm. We then 'unwrap' the geodesic distances onto a two dimensional surface; the

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

	Points	Radius	RMS err	Struct. err	Time
Our method	642	1.00	0.0030	9×10^{-6}	7.2s
Elad et al		-	0.0029	3×10^{-5}	26.9s
Kernel		-	0.39	1×10^{-5}	0.8s

TABLE I
THE PERFORMANCE OF EMBEDDING METHODS ON THE SPHERE TEXTURING PROBLEM.

	Points	Radius	RMS err	Struct. err	Time
Our method	1016	0.78	0.21	0.11	31s
Elad et al		-	0.21	0.11	206s
Kernel		-	0.24	0.14	4.3s

TABLE II
THE PERFORMANCE OF EMBEDDING METHODS ON THE STANFORD BUNNY TEXTURING PROBLEM.

sphere S^2 for the spherical embeddings and the plane R^2 for the kernel embedding. This embedding is used to map a texture back onto the surface of the object. The texture is defined on the surface of a sphere for the spherical embeddings and on the plane for kernel embedding. For visualization purposes only, we subdivide the mesh after embedding to enable us to view the texture in high resolution. Any distortions in the embedding are revealed as distortions in the surface texture map.

The first model is a simple test case of the sphere. Figure 1 shows the results of texture-mapping the surface with a triangular texture. These results are summarized in Table I. Both spherical embedding methods produce near-perfect embeddings of the surface. Since our method is not based on optimization, it is considerably quicker than the method of Elad et al [12]. However, these times are only indicative as neither algorithm was optimized for speed[1]. While the kernel method is much quicker, there is naturally considerable distortion in mapping the sphere to a plane.

The second model is the Stanford bunny. This model is subsampled to 1016 vertices and 2000 faces. The embedding results are shown in Figure 2 and Table II. Again, the spherical embedding methods produce very similar results. However, our method finds a radius of curvature of 0.78 which is considerably different from 1. Note the distortion of the texture around the ears for the kernel methods, which is not present in the other methods.

B. Embedding Time-Warping Functions

We now turn our attention to a computer vision problem to assess the ability of our spherical embedding method to discover low-dimensional embeddings. As we discussed in Section II, there are a wide range of problems where data exists on a spherical manifold. Examples include document analysis, comparison of spectra, histogram distances and time-warping functions. Here we take an example from Veeraraghavan et al [33] based on sets of time-warping functions $\gamma(t)$.

[1]The methods were implemented in Matlab on a Intel Core2 Duo 3GHz machine

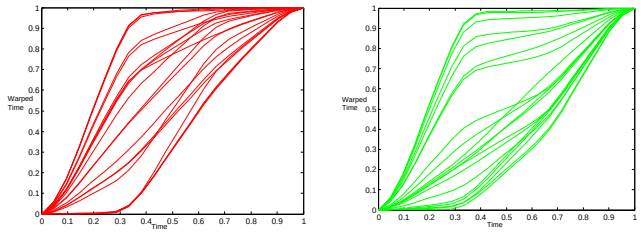

Fig. 3. The two classes and distributions of time-warp functions.

Under the reparameterisation $\psi(t) = \sqrt{\dot{\gamma}(t)}$ the square-root density form lies on a hypersphere of dimension equal to the number of time-samples of $\psi(t)$. Veeraraghavan et al demonstrate the advantage of modelling the prior distributions on a hypersphere.

We generate some time-warping functions in two classes. Each class also has one free parameter which we vary to generate different examples, and a small amount of uniform noise on the class parameters. We sample 50 points from each time warping function $\gamma(t)$ and re-parameterize it into the form $\psi(t)$. The data therefore lies on a 50-dimensional hypersphere, but the underlying structure is approximately two-dimensional. Samples of the time-warping functions for the two classes are shown in Figure 3.

Using spherical embedding we can recover a two-dimensional embedding of the points. We begin by computing a distance matrix for the points and rescaling so that the mean distance is one. This step is not strictly necessary for this problem as we can directly compute the matrix $\mathbf{Z}$, but we use this method for consistency with the other experiments. Using the embedding procedure described in Section V, we recover a 2-dimensional spherical embedding as expected. This is shown in Figure 4 and clearly shows the structure of the data. The embedding radius is $r = 1.96$, the RMS embedding error is 4×10^{-3} and the structural error is 10^{-4}, illustrating that the method accurately recovers the low-dimensional embedding.

C. Robustness to Noise

We now turn our attention to more challenging problems, where the embedding manifold is more than two-dimensional and noise and distortions are present. The methods based on spherical-polar coordinates are difficult to use in this situation [12], [7] as it becomes increasingly complex to model the coordinate system in a larger number of dimensions. Although our method finds the embedding exactly when the points lie on the hypothesised surface, in realistic situations there is invariably noise in the measurements. In order to assess the performance of our methods on noisy data, we generate synthetic data with controlled levels of noise as follows. We begin by generating points on the surface of a sphere (or hyperboloid). In this experiment the sphere is embedded in 50-dimensional space and 50 points are generated. We then construct the distance matrix for the points using the geodesic distance on the sphere. These distances are then perturbed by chi-squared noise on the squared distances of varying amounts; this preserves the positivity of the distances, which would

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

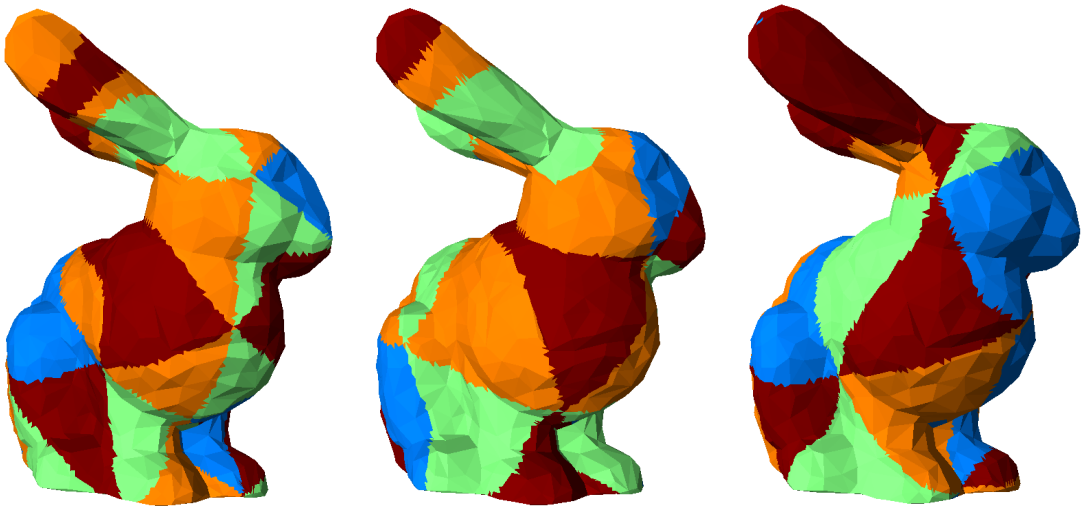

Fig. 2. Texture mapping of the Stanford bunny. Left: using our method, middle: the method of Elad et al, and right: kernel embedding. The spherical embeddings produce similar results. Note the distortion around the ears in the kernel embedding.

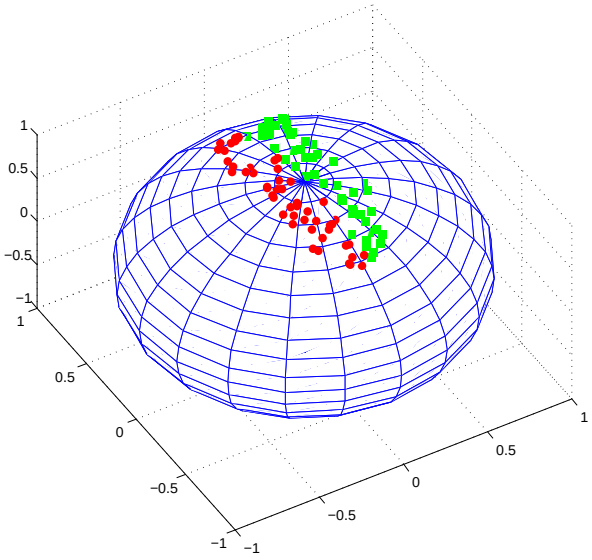

Fig. 4. Spherical embedding of the time-warp functions in two dimension, when encoded in square-root density form.

not be true of Gaussian noise. The same noise is applied symmetrically to $\mathbf{D}$ to maintain symmetry of the distances. We finally apply our embedding methods to the noisy distances. The results are shown in Figure 5 for the spherical space and Figure 6 for the hyperbolic space. The errors are computed between the noisy distance matrix and its embedding (i.e. they are the errors cause by the embedding process only). For comparison, we include the difference between the noisy distance matrix and the original(with no noise).

It is clear from Figure 5 that the spherical embedding is effective even in the presence of large amounts of noise. At all noise levels, the distortion of the spherical embedding is less than that caused by the kernel embedding. The spherical embedding also shows a remarkable ability to preserve the neighborhood order (structural error). This is also apparent, although to a lesser extent, in real-world datasets (section VII-D. Optimization of the spherical embedding produces good embedding results but increases the structural error significantly.

The hyperbolic embedding (Figure 6) is affected more significantly by noise - the performance is good for low noise levels but at moderate to high noise the errors are similar to that of the kernel embedding. Optimization does give a significant improvement in embedding accuracy. It appears to be more difficult to accurately locate the correct radius of curvature for the hyperbolic space in the presence of noise.

D. Dissimilarity-based datasets

Finally, we use some data from a selection of similarity-based pattern recognition problems [22], [1], [34], [18]. These are a subset of the data analyzed under the SIMBAD project (simbad-fp7.eu), and more details of the datasets are available in the relevant project report [11]. They are selected on the basis that they have small radius-of-curvature under the spherical model, and therefore are significantly non-Euclidean under those models. The data is based on classification problems, and so as well as showing measurements of distortion, we also calculate the k-nearest-neighbour classification error rate. We choose this classifier as it can be operated in both Euclidean and non-Euclidean spaces as it uses only dissimilarity data. Information about the datasets is given in Table III. Here we provide a brief description of each data set.

DelftGestures: This dataset consists of the dissimilarities computed from a set of gestures in a sign-language study. They are measured by two video cameras observing the positions the two hands in 75 repetitions of 20 different signs. The dissimilarities result from a dynamic time warping procedure [18].

FlowCyto-1: This dissimilarity dataset is based on 612 FL3-A DNA flowcytometer histograms from breast cancer tissues in 256-bin resolution. The initial data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

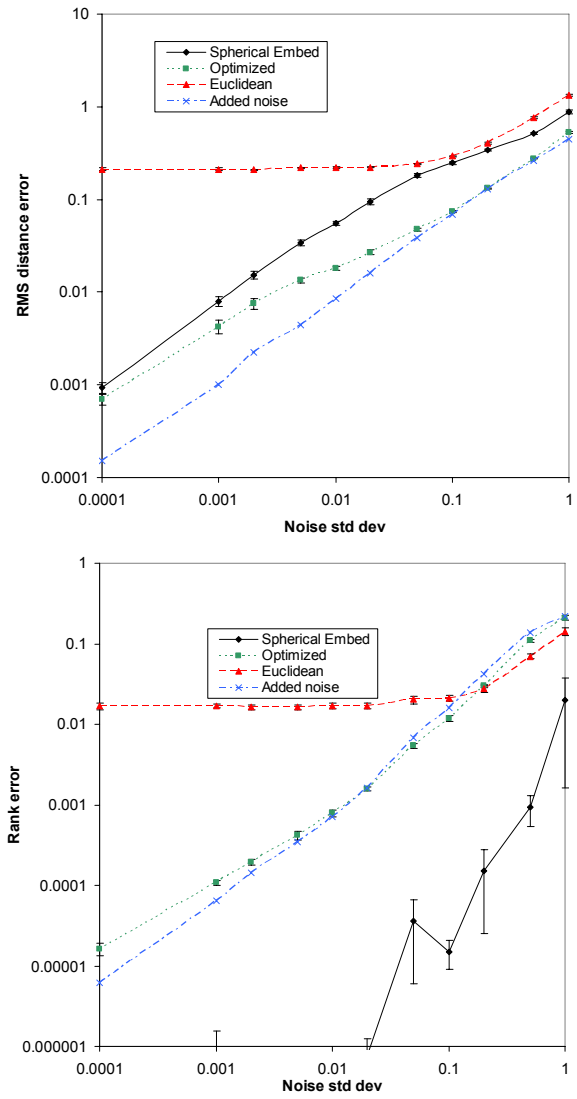

Fig. 5. Reconstruction of noisy distances via a range of different embedding methods.

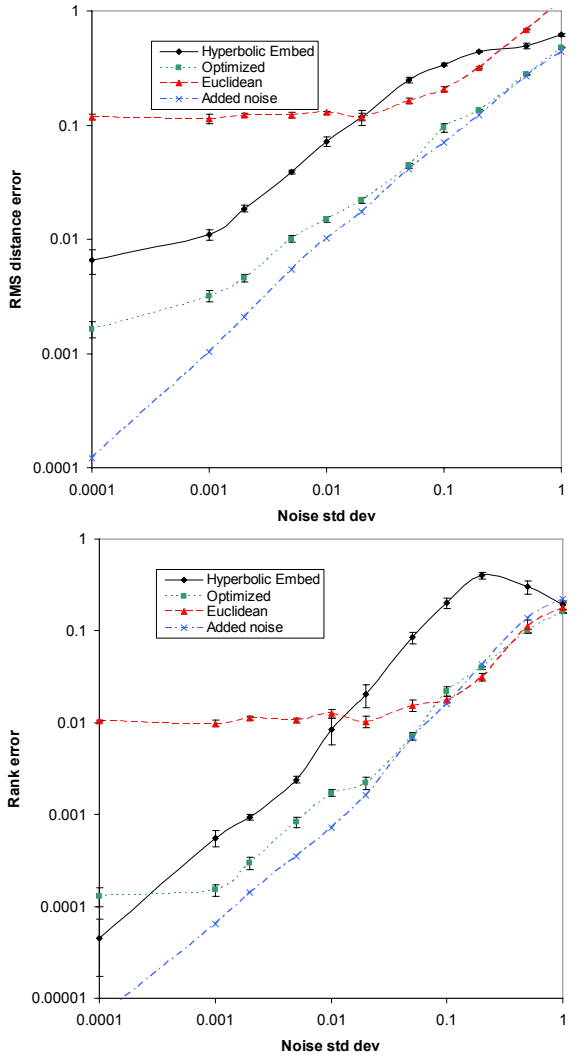

Fig. 6. Reconstruction of noisy distances via a range of different embedding methods.

in Heerlen, The Netherlands, during 2000-2004. Histograms are labeled in 3 classes: aneu- ploid (335 patients), diploid (131) and tetraploid (146). Dissimilarities between normalized histograms are computed using the L1 norm, correcting for possible different calibration factors.

Chickenpieces-25-45: This is one of the chickenpieces dissimilarity matrices as made available by Bunke et.al. [5] Every entry is a weighted string edit distance between two strings representing the contours of 2D blobs. Contours are approximated by vectors of length 25. Angles between vectors are used as replacement costs. The costs for insertion and deletion are 45.

Catcortex: The cat-cortex data set is provided as a 65x65 dissimilarity matrix describing the connection strengths between 65 cortical areas of a cat from four regions (classes): auditory (A), frontolimbic (F), somatosensory (S) and visual (V). The data was collected by Scannell et al [26]. The dissimilarity values are measured on an ordinal scale.

CoilYork: These distances represent the approximate graph edit distances between a set of graphs derived from views of four objects in the COIL image database. The graphs are the Delaunay graphs created from the corner feature points in the images [35]. The distances are computed with the graduated assignment method of Gold and Rangarajan [14].

Newsgroups: This is a small part of the commonly used 20 Newsgroups data. A non-metric correlation measure for messages from four classes of newsgroups, 'comp.*', 'rec.*','sci.*' and 'talk.*' are computed on the occurrence for 100 words across 16242 postings.

In a realistic scenario for such data, we might use a manifold learning technique to embed the data into a small number of dimensions. Here we use 10 embedding dimensions for all our techniques. We compare the spherical embedding (with and without optimization) to kernel embedding, ISOMAP [31] and Laplacian Eigenmaps [2]. As well as the RMS embedding error and structural error as described earlier, we also include the k-NN leave-one-out cross-validation error for these classification problems. This is computed from the distances in the

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

Data (size)		Radius	RMS Err	Struct. Err	kNN Erate%(k)	Optimal NG
DelftGestures	spherical	0.661	0.102	0.051	8.5(5)± 0.7	–
(1500)	optimized	0.745	**0.078**	0.030	8.7(5)± 0.7	–
	kernel	–	0.088	**0.028**	**7.5**(3)± 0.7	–
	ISOMAP	–	0.088	0.031	9.9(20)± 0.8	1320
	Lap. Eigenmap	–	–	0.474	25(5)± 1.1	526
FlowCyto-1	spherical	0.847	0.085	**0.031**	33.0(28)± 1.9	–
(612)	optimized	0.865	**0.079**	0.033	32.1(12)± 1.9	–
	kernel	–	0.156	0.054	32.0(11)± 1.9	–
	ISOMAP	–	0.089	0.032	**31.8**(13)± 1.9	600
	Lap. Eigenmap	–	–	0.262	33.8(79)± 1.9	235
Chickenpieces-25-45	spherical	0.749	0.160	0.074	4.0(4)± 0.9	–
(446)	optimized	0.792	**0.117**	**0.073**	**3.8**(1)± 0.9	–
	kernel	–	0.206	0.085	4.5(5)± 1.0	–
	ISOMAP	–	0.162	0.075	4.9(3)± 1.0	275
	Lap. Eigenmap	–	–	0.413	14(3)± 1.6	189
Catcortex	spherical	0.365	0.458	0.350	5.1(3)± 2.7	–
(65)	optimized	0.702	**0.128**	**0.245**	**3.1**(8)± 2.1	–
	kernel	–	0.163	0.318	4.6(5)± 2.6	–
	ISOMAP	–	0.156	0.318	6.2(2)± 3.0	53
	Lap. Eigenmap	–	–	0.411	18.5(7)± 4.8	20
CoilYork	spherical	0.802	0.240	**0.138**	**32.0**(7)± 2.7	–
(288)	optimized	0.900	**0.146**	0.163	36.8(8)± 2.8	–
	kernel	–	0.331	0.158	36.9(18)± 2.8	–
	ISOMAP	–	0.194	0.196	44.7(6)± 2.9	166
	Lap. Eigenmap	–	–	0.531	44.3(9)± 2.9	128
Newsgroups	spherical	0.331	0.503	0.633	**20.8**(35)± 1.7	–
(288)	optimized	0.622	**0.230**	0.628	25.2(76)± 1.8	–
	kernel	–	0.474	0.663	26.0(31)± 1.8	–
	ISOMAP	–	0.243	**0.571**	21.9(3)± 1.7	130
	Lap. Eigenmap	–	–	0.580	30.3(13)± 1.9	68

TABLE IV
SPHERICAL EMBEDDINGS OF SOME SIMILARITY-BASED DATASETS, COMPARED TO THE KERNEL EMBEDDING.

Dataset	Size	C	NEF	Asym	TVF	kNN(k)
DelftGestures	1500	20	0.308	0	4×10^{-6}	3.1(29)
FlowCyto-1	612	3	0.275	0	1×10^{-3}	32(30)
Chicken	446	5	0.320	0.063	2×10^{-5}	4.3(3)
Catcortex	65	4	0.208	0	1×10^{-3}	9(61)
CoilYork	288	4	0.027	0	0	23(13)
Newsgroups	600	4	0.202	0	2×10^{-5}	22(1)

TABLE III
DATASETS USED IN THE SIMILARITY BASED EXPERIMENTS. SEE TEXT FOR MORE DETAILS

embedding space. In all cases we choose k to achieve the best classification performance. There is an uncertainty associated with the LOO error-rate due to the limited number of testing samples, which is given as one standard deviation errors in the table.

Firstly we note that the kernel embedding has no free parameters. Spherical embedding has one parameter, the radius, which is determined from the data as discussed earlier. ISOMAP and Laplacian Eigenmap also have a free parameter, the order of the nearest-neighbor graph to use. For ISOMAP, we choose this parameter to achieve the lowest RMS embedding error. Laplacian Eigenmap is not a distance preserving embedding, and so it does not make sense to compare the RMS embedding errors. For LEM, we choose the graph order to minimize the structural error.

The results are summarized in Table IV. The values are shown in bold for the best result. The goal of the spherical embedding is to minimise the embedding error of the points and we can see that in every case the optimized spherical embedding achieves the lowest RMS embedding error. The spherical embedding (either optimized or unoptimized) achieves the lowest structural error in four out of the six cases. The exceptions is the DelftGestures data where kernel embedding is best, but very similar to optimized spherical embedding (0.028 vs. 0.030) and the Newsgroups data, where LEM performs better. The classification rates are best with spherical embedding again in 4 out of 6 cases, the exceptions being FlowCyto-1 where the results are all very similar, and DelftGestures where kernel embedding performs particularly well.

E. Hyperbolic Embedding of Trees

We also analyzed our similarity-based datasets (discussed in the previous section) for hyperbolic-like examples by looking for those with significant curvatures under the hyperbolic model. When embedding in 10 dimensions, there were no such examples. We believe that this is because hyperbolic space is unbounded and the volume available increases exponentially, as opposed to polynomially for Euclidean and spherical space. Since (dis)similarities are usually bounded, the spherical space is a more natural setting. However, hierarchical data such as trees and complex networks may naturally be embedded in hyperbolic spaces [8] when the number of elements at each level of the hierarchy increases exponentially. From

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

b(size)		Radius	RMS Err	Str. Err	Opt. NG
2(63)	hyperbolic	0.998	0.0146	0.037	–
	optimized	0.989	**0.0041**	**0.012**	–
	kernel	–	0.0581	0.032	–
	ISOMAP	–	0.0581	0.032	350
	LapE	–	–	0.343	28
3(364)	hyperbolic	0.657	0.0033	**0.049**	–
	optimized	0.656	**0.0010**	**0.049**	–
	kernel	–	0.0354	0.050	–
	ISOMAP	–	0.0354	0.050	350
	LapE	–	–	0.514	202
4(1365)	hyperbolic	0.441	0.0032	**0.097**	–
	optimized	0.441	**0.0005**	**0.097**	–
	kernel	–	0.0244	**0.097**	–
	ISOMAP	–	0.0244	**0.097**	1300
	LapE	–	–	0.555	400

TABLE V
EMBEDDINGS OF TREES WITH DEPTH 5 AND VARIOUS BRANCHING FACTORS.

the embedding method in Krioukov et al [8] we can derive a simple distance heuristic for the nodes of a tree which approximates the hyperbolic disk model.

We take a tree with branching factor b and place the root node at the centre of the disk. The depth in the tree of node i is r_i. We define the angular distance between nodes i and j as

$$\Delta\theta_{ij} = \begin{cases} \frac{\pi}{4} b^{-\frac{r_i+r_j-p_{ij}}{2}} & \text{If } i,j \text{ are directly related} \\ \frac{2\pi(b+1)}{3b} b^{-\frac{r_i+r_j-p_{ij}}{2}} & \text{If } i,j \text{ otherwise} \end{cases}$$

These distances are the expected angular separations of the nodes. The hyperbolic distance d_{ij} between nodes is then

$$\begin{aligned} \cosh(d_{ij}\ln b) &= \cosh(r_i \ln b) + \cosh(r_j \ln b) \\ &\quad - \sinh(r_i \ln b)\sinh(r_j \ln b)\cos\Delta\theta_{ij} \end{aligned}$$

Armed with these distances, we can use hyperbolic embedding to recover an embedding of the tree. The results are summarized in Table V for trees of depth 5 and varying branching factors. In all cases, the hyperbolic embedding gives excellent embeddings with low distortion and close to the theoretical curvature $\propto \ln b$. Optimization gives near-perfect embeddings in terms of the RMS error which are not achievable by the other methods.

VIII. CONCLUSION

Spaces of constant-curvature offer a useful alternative for the embedding of non-Euclidean datasets. This allows intrinsically non-flat geometry between objects and may be a better description for many datasets.

In this paper we have presented efficient methods of embedding points into spherical and hyperbolic spaces using their distance matrices. This simple method is based on the eigendecomposition of a similarity matrix which determines the curvature, followed by a correction to project into constant-curvature space. We also developed an optimization procedure for improving the accuracy of the embedding for more difficult datasets which utilized the exponential map to transform the problem into an optimization in tangent space.

Our results on synthetic and real data show that the spherical embedding performs well under noisy conditions and can deliver low-distortion embeddings for a wide variety of datasets. Hyperbolic-like data seems to be much less common (at least in our datasets) and is more difficult to accurately embed. Nevertheless, in low-noise cases the hyperbolic space can also be used to accurately embed non-Euclidean dissimilarity data. In all the data tested here (with significant non-Euclidean behaviour) the spherical embedding delivers the lowest RMS distortion error.

While accurate embedding is our goal here, it is natural to want to apply pattern recognition techniques to the embedded data. In most cases, the spherical embeddings give a competitive kNN classification error rate. However, the key advantage of these spaces is that they are metric, and so it should be possible to apply geometric techniques (such as LDA or the SVM) in these spaces. However, many methods currently rely, either explicitly or implicitly, on an underlying kernel space which is Euclidean. We believe that much more work needs to be done in the future on applying such techniques in non-flat spaces.

REFERENCES

[1] G. Andreu, A. Crespo, and J.M. Valiente. Selecting the toroidal self-organizing feature maps (tsofm) best organized to object recognition. In *ICNN'97*, pages 1341–1346, 1997.
[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
[3] M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2003 Conference (NIPS)*, pages 961–968. MIT Press., 2003.
[4] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Expression-invariant face recognition via spherical embedding. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–756–9, 2005.
[5] H. Bunke and U. Buhler. Applications of approximate string matching to 2d shape recognition. *Pattern Recognition*, 26:1797–1812, 1993.
[6] M. A. A. Cox and T. F. Cox. Multidimensional scaling. In *Handbook of Data Visualization*, pages 315–347. Springer Handbooks of Computational Statistics, 2008.
[7] T. F. Cox and M. A. A. Cox. Multidimensional scaling on a sphere. *Communications in Statistics - Theory and Methods*, 20(9):2943–2953, 1991.
[8] M. Kitsak A. Vahdat M. Boguna D. Krioukov, F. Papadopoulos. Hyperbolic geometry of complex networks. *Physical Review E*, 82:036106, 2010.
[9] J. De Leeuw. Applications of convex analysis to multidimensional scaling. In J. R. Barra, F. Brodeau, G. Romier, and B. van Cutsem, editors, *Recent Developments in Statistics*, pages 133–145, 1977.
[10] J. De Leeuw and W. J. Heiser. Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah, editor, *Multivariate Analysis*, pages 501–522, 1980.
[11] R. P.W. Duin and E. Pękalska. Datasets and tools for dissimilarity analysis in pattern recognition. Technical report, SIMBAD Deliverable 3.3, http://www.simbad-fp7.eu/images/D3.3_Datasets_and_tools_for_dissimilarity_analysis_in_pattern_recognition.pdf, 2009.
[12] A. Elad, Y. Keller, and R. Kimmel. Texture mapping via spherical multidimensional scaling. In *Scale-Space 2005*, volume 3459 of *LNCS*, pages 443–455, 2005.
[13] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, 2004.
[14] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:377–388, 1996.
[15] L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17:575–582, 1984.

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2014.2316836

[16] L. Hubert, P. Arabie, and J. Meulman. Linear and circular unidimensional scaling for symmetric proximity matrices. *British Journal of Mathematical & Statistical Psychology*, 50:253–284, 1997.
[17] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.
[18] J. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders. Sign language recognition by combining statistical DTW and independent classfication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:2040–2046, 2008.
[19] H. Lindman and T. Caelli. Constant curvature Riemannian scaling. *Journal of Mathematical Psychology*, 17:89–109, 1978.
[20] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal of Computing*, 16:647–668, 1987.
[21] Elzbieta Pekalska and Robert P. W. Duin. *The dissimilarity representation for pattern recognition*. World Scientific, 2005.
[22] Elzbieta Pekalska, Artsiom Harol, Robert P. W. Duin, Barbara Spillmann, and Horst Bunke. Non-Euclidean or non-metric measures can be informative. In *SSPR/SPR*, pages 871–880, 2006.
[23] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.
[24] A. Robles-Kelly and E. R. Hancock. A Riemannian approach to graph embedding. *Pattern Recognition*, 40:1042–1056, 2007.
[25] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
[26] J. Scannell, C. Blakemore, and M. Young. Analysis of connectivity in the cat cerebral cortex. *Journal of Neuroscience*, 15:1463–1483, 1995.
[27] I.J. Schoenberg. On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in hilbert space. *Annals of Mathematics*, 38:787–797, 1937.
[28] Y. Shavitt and T. Tankel. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Transactions on Networking*, 16:25–36, 2008.
[29] R. N. Shepard. Representation of structure in similarity data: Problems and prospects. *Pyschometrika*, 39:37300421, 1974.
[30] A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.
[31] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
[32] 1958. Torgerson, W.S. *Theory and methods of scaling.* Wiley, New York, 1958.
[33] A. Veeraraghavan, A. Srivastava, A.K. Roy-Chowdhury, and R. Chellappa. Rate-invariant recognition of humans and their activities. *IEEE Transactions on Image Processing*, 18(6):1326 –1339, june 2009.
[34] B. Xiao and E. R. Hancock. Geometric characterisation of graphs. *ICIAP 2005*, LNCS 3617:471–478, 2005.
[35] B. Xiao and E. R. Hancock. Geometric characterisation of graphs. In *ICIAP 2005, LNCS 3617*, pages 471–478, 2005.
[36] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal on Scientific Computing.*, 26(1):313–338, 2004.

Richard Wilson received the BA degree in Physics from the University of Oxford in 1992. In 1996 he completed a PhD degree at the University of York in the area of pattern recognition. From 1996 to 1998 he worked as a Research Associate at the University of York. After a period of postdoctoral research, he was awarded an EPSRC Advanced Research Fellowship in 1998. In 2003 he took up a lecturing post and he is now a Professor in the Department of Computer Science at the University of York. He has published more than 150 papers in international journals and refereed conferences. He received an outstanding paper award in the 1997 Pattern Recognition Society awards and has won the best paper prize in ACCV 2002. He is a member of the editorial board of the journal Pattern Recognition, a Fellow of the International Association for Pattern Recognition and a senior member of IEEE. His research interests are in statistical and structural pattern recognition, graph methods for computer vision, high-level vision and scene understanding.

Edwin Hancock received the B.Sc., Ph.D., and D.Sc. degrees from the University of Durham, Durham, U.K. He is now a Professor of computer vision in the Department of Computer Science, University of York, York, U.K. He has published nearly 140 journal articles and 500 conference papers. Prof. Hancock was the recipient of a Royal Society Wolfson Research Merit Award in 2009. He has been a member of the editorial board of the IEEE TPAMI, Pattern Recognition, CVIU, and Image and Vision Computing. His awards include the Pattern Recognition Society Medal in 1991, outstanding paper awards from the Pattern Recognition Journal in 1997, and the best conference best paper awards from the Computer Analysis of Images and Patterns Conference in 2001, the Asian Conference on Computer Vision in 2002, the International Conference on Pattern Recognition (ICPR) in 2006, British Machine Vision Conference (BMVC) in 2007, and the International Conference on Image Analysis and Processing in 2009. He is a Fellow of the International Association for Pattern Recognition, the Institute of Physics, the Institute of Engineering and Technology, and the British Computer Society.

Elżbieta Pękalska received the MSc degree in computer science from the University of Wroclaw, Poland, in 1996 and the PhD degree (cum laude) in computer science from Delft University of Technology, Delft, The Netherlands, in 2005. During 1998-2004, she was with Delft University of Technology, The Netherlands, where she worked on both fundamental and applied projects in pattern recognition. Until 2012 she was an engineering and physical sciences research council fellow at the University of Manchester, United Kingdom. She is engaged in learning processes and learning strategies, as well as in the integration of bottom-up and top-down approaches, which not only includes intelligent learning from data and sensors, but also human learning on their personal development paths. She is the author or coauthor of more than 40 publications, including a book, journal articles, and international conference papers. Her current research interests focus on the issues of representation, generalization, combining paradigms, and the use of kernels and proximity in the learning from examples. She is also involved in the understanding of brain research, neuroscience, and psychology.

Robert P.W. Duin received the Ph.D. degree in applied physics from the Delft University of Technology, Delft, The Netherlands, in 1978, for a thesis on the accuracy of statistical pattern recognizers. In his research, he included various aspects of sensors, image processing, parallel computing, the automatic interpretation of measurements, learning systems, and classifiers. Currently, he is an Associate Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. His present research is in the design, evaluation, and application of algorithms that learn from examples. Recently, he started to investigate alternative object representations for classification and became thereby interested in dissimilarity-based pattern recognition and in the possibilities to learn domain descriptions.

Dr. Duin is a former Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence. He is a Fellow of the IAPR. In August 2006, he received the Pierre Devijver Award for his contributions to statistical pattern recognition.

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.