Fast Manifold Learning Based on Riemannian Normal Coordinates

Anders Brun[1,3], Carl-Fredrik Westin[3], Magnus Herberthson[2], and Hans Knutsson[1]

[1] Department of Biomedical Engineering, Linköpings Universitet, Linköping, Sweden
{andbr,knutte}@imt.liu.se
[2] Department of Mathematics, Linköpings universitet, Linköping, Sweden
maher@mai.liu.se
[3] Laboratory of Mathematics in Imaging, Harvard Medical School, Boston, MA, USA
westin@bwh.harvard.edu

Abstract. We present a novel method for manifold learning, i.e. identification of the low-dimensional manifold-like structure present in a set of data points in a possibly high-dimensional space. The main idea is derived from the concept of Riemannian normal coordinates. This coordinate system is in a way a generalization of Cartesian coordinates in Euclidean space. We translate this idea to a cloud of data points in order to perform dimension reduction. Our implementation currently uses Dijkstra's algorithm for shortest paths in graphs and some basic concepts from differential geometry. We expect this approach to open up new possibilities for analysis of e.g. shape in medical imaging and signal processing of manifold-valued signals, where the coordinate system is "learned" from experimental high-dimensional data rather than defined analytically using e.g. models based on Lie-groups.

1 Introduction

A manifold can be seen as a generalization of a surface to higher dimensions. Locally a manifold looks like a Euclidean space, $\mathbb{R}^N$, but on a global scale it may be curved and/or compact, like a sphere or a torus. A manifold with a metric tensor defined at each point is called a Riemannian manifold.

Recent developments in so called manifold learning has opened up new perspectives in non-linear data analysis. Classical methods such as Principal Components Analysis (PCA, a.k.a. the Karhunen-Loeve transform) and Multidimensional Scaling (MDS) efficiently finds important linear subspaces in a set of data points. Methods within the field of manifold learning are however able to identify non-linear relations as well. In this paper we present a new tool for data analysis of this kind, based on the concept of Riemannian normal coordinates.

Manifold learning has become an established field of research, Kohonen's Self Organizing Maps (SOM) [5] being an important early example. Characteristic for the newest generation of manifold learning techniques is efficiency and global convergence, in particular many of them are based on the solution of very large eigenvalue problems. This include for instance the recent Kernel PCA [8], Locally Linear Embedding [7], ISOMap [11], Laplacian Eigenmaps [1] and Hessian Eigenmaps [2].

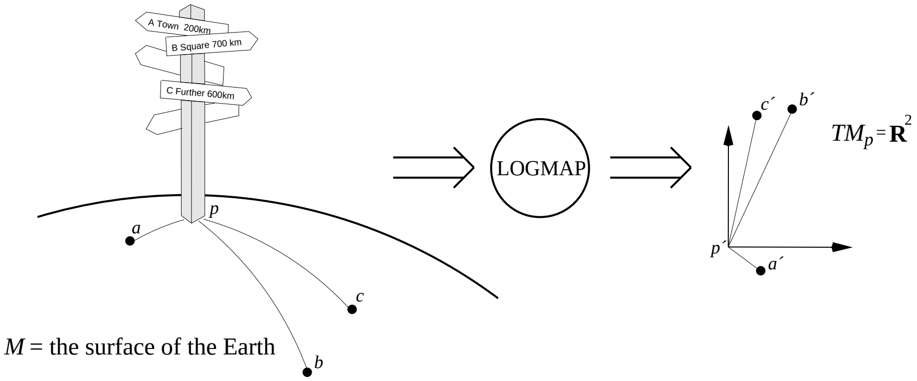

Fig. 1. Traveling along a geodesic, starting at a specific location in a specific direction, will eventually take you to any place on the surface of the Earth. Riemannian normal coordinates captures this information, mapping points on the sphere to $\mathbb{R}^2$ in a way that direction and geodesic distance from the origin to any point is preserved. Riemannian normal coordinates are therefore quite natural to use for navigation on a manifold, at least in the vicinity of a point. Also note that geodesics on a manifold M (left) are mapped to lines in Riemannian normal coordinates (right).

Manifolds arise in data for instance when a set of high-dimensional data points can be modeled in a continuous way using only a few variables. A typical example is a set of images of a 3-D object. Each image may be represented as a very high-dimensional vector, which depends on the scene and a few parameters such as relative camera orientation, camera position and lighting conditions. Camera orientation itself is a good example of a non-linear manifold. The manifold of orientations, $SO(3)$, can be represented by the set of all rotation matrices. While the manifold-valued parameter space is equivariant to important features of the data, namely camera- and lighting information, it should also be invariant to uninteresting things such as noise from the image sensors.

In the following sections we present a novel technique for manifold learning based on the concept of Riemannian normal coordinates. We have translated this technique from its original setting in differential geometry, to the task of mapping a set of experimental high-dimensional data points, with a manifold-like structure, to a low-dimensional space. An intuitive explanation of Riemannian normal coordinates is given in figure 1. They contain information about the direction and distance from a specific point on a manifold to other nearby points. The usefulness of such information for navigation is obvious, not only for navigating on the Earth, but also for creating user interfaces to navigate in manifold-valued data in general. The Riemannian normal coordinates are also closely related to geodesics and the exponential and logarithmic maps of Lie-groups, which have been used recently for the analysis of shape in medical images [4] and to perform time-dependent signal processing of orientation data [6].

2 Theory

In this section we briefly review some basic concepts of differential geometry necessary to understand the method we propose.

To each point p on a manifold M there is a associated tangent space, T_pM, consisting of a Euclidean space tangential to M at p. Derivatives at p of smooth curves passing through a point p belongs to T_pM.

A special class of curves defined on Riemannian manifolds are the geodesics, i.e. length minimizing curves on M. These define a metric $d(x, y)$ on a manifold derived from the length of a geodesic passing through x and y.

The Riemannian exponential map, $\exp(v) \in M$, $v \in T_pM$, is a function which maps points in the tangent space of p, to points on M. If $H(t)$ is the unique geodesic, starting at p with velocity v, then $\exp(v) = H(1)$. Intuitively this can be thought of as walking with constant velocity in particular direction on the manifold, from a point p, during one time unit. This mapping is one-to-one in a neighborhood of p and its inverse is the log map.

The set of points on M for which there exists more than one shortest path from p is called the *cut locus* of p. The cut locus of a point on a sphere is for instance its antipodal point. Some manifolds, such as $\mathbb{R}^2$, lack a cut locus. Other manifolds, such as the torus, have a quite complex looking cut locus.

Given a point p and an orthonormal basis $\{\hat{\mathbf{e}}_i\}$ for the tangent space T_pM, a Riemannian normal coordinate system is provided by the exponential mapping. A point $x \in M$ gets the coordinate $(x^1, \ldots, x^N)$ if $x = \exp(x^i \hat{\mathbf{e}}_i)$.

The gradient of a scalar function f is a dual vector field which components are simply the partial derivatives (in the induced basis).

3 Method

Given a basis point p from a data set X and an orthonormal basis of the tangent space at p to a thought manifold M, we would like to, via the log map into T_pM, express all data points $x \in X$ using Riemannian normal coordinates. Due to the properties of Riemannian normal coordinates, this is equivalent to measuring the distance and direction from p to every other point in the data set. We choose to call this framework LOGMAP:

1. From a set of data points, X, sampled from a manifold M, choose a base point $p \in X$.
2. To determine the dimension of M, select a ball $B(p)$ of the K closest points around p. Then perform standard PCA in the ambient space for $B(p)$. This will give us T_pM, with $\dim T_pM = N$, where we choose any suitable ON-basis $\{\hat{\mathbf{e}}_i\}$. All $y \in B(p)$ are mapped to T_pM by projection on $\{\hat{\mathbf{e}}_i\}$ in the ambient space. This is the Ψ-mapping in figure 2.
3. Approximate distances on M. In the current implementation we do this by defining a weighted undirected graph, with each node corresponding to a data point and with edges connecting each node to its L closest neighbors. Let the weights of these

edges be defined by the Euclidean distance between data points in the ambient space. We then use Dijkstra's algorithm for finding shortest paths in this graph, to approximate the geodesic distances in M. This gives estimates of $d(x, y)$ for all $(x, y) \in X \times B(p)$.

4. To calculate the direction from p to every point $x \in X$, estimate $\mathbf{g} = \sum g^i \hat{\mathbf{e}}_i = \nabla_y d^2(x, y)|_{y=p}$ numerically, using the values obtained in the previous step. While we only have values of $d^2(x, y)$ for $y \in B(p)$, we must interpolate this function in T_pM, e.g. using a second order polynomial, in order to calculate the partial derivatives at $\Psi(p)$.
5. Estimates of Riemannian normal coordinates for a point x are then obtained as $x^i = d(x, p)\frac{g^i}{|\mathbf{g}|}$.

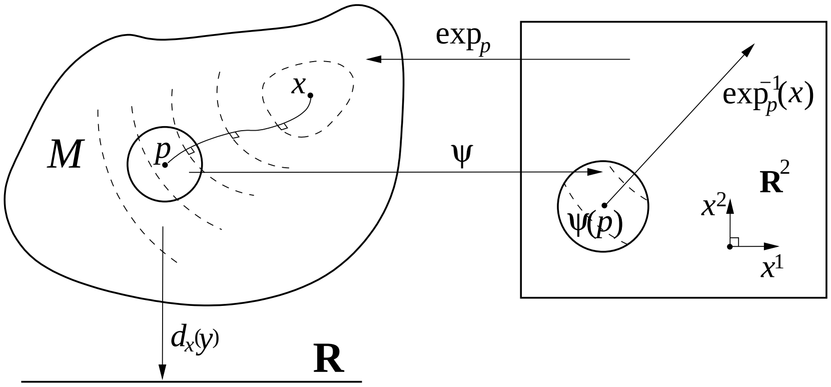

Fig. 2. A schematic illustration of a geodesic from x to p in a manifold M. Dashed curves correspond to iso-levels of $d_x^2(y) = d^2(x, y)$. These iso-curves are perpendicular to every geodesic passing through x. The ball around p and the mapping Ψ defines a chart that maps a part of M to $\mathbb{R}^2$. The domain of $\exp$ is actually the tangent space of M at p, and it is natural to identify vectors in $\mathbb{R}^2$ with TM_p.

In step 4) above, the numerical calculation of the gradient at p uses the squared distance function. The reason for not just taking the gradient at p of the plain distance function from x, which is known to point in the direction of the geodesic connecting p and x, is that it is not smooth for $p \approx x$. Using the square of the distance function, which is much easier to interpolate, solves this problem while giving a gradient in the same direction. However, when x is close to the cut locus of p, even the squared distance function becomes non-smooth. In the experiments shown in the next section, we have actually used a slightly more robust scheme to estimate the gradient for points close to the cut locus. This was done by using the RANSAC algorithm [3], to select points close to p consistent with a second order polynomial model of the squared distance function.

4 Experiments

The LOGMAP method was evaluated using Matlab. The most critical part of the algorithm, the calculation of shortest paths, was borrowed from the ISOMAP implementation of Dijkstra's shortest paths [11]. In the LOGMAP implementation, the selection of p was made interactively by the click on the mouse and the resulting log map was calculated almost in real time.

Three experiments on synthetic data are presented here to illustrate the behavior of the algorithm. In each of the experiments we have assumed knowledge of how to choose L, the number of neighbors for building the graph, and K, which determines the size of the neighborhood used for dimension estimation and later the estimation of gradients. It is important to point out that selection of these parameters is actually non-trivial for many data sets, e.g. when noise is present. We will not go further into the details of choosing these constants in this paper however.

4.1 The Swiss Roll

In the first experiment we use the "Swiss roll" data set, consisting of points sampled from a 2-D manifold, embedded in $\mathbb{R}^3$, which looks like a roll of Swiss cheese. It has been used before to illustrate methods for manifold learning, see e.g. [11, 7], and we include it mainly as a benchmark. A set of 2000 points from this data set were used in the experiment and the results are presented in figure 3. The experiment shows that the LOGMAP method correctly unfolds the roll and maps it to Riemannian normal coordinates in $\mathbb{R}^2$.

It is important to note that the resulting mapping in the Swiss roll example is more or less isometric, which is expected for simple flat manifolds. This is similar to the behavior of ISOMAP. On the other hand, both ISOMAP and LOGMAP would fail to produce isometric embeddings if we would introduce a hole in the Swiss roll data set. This particular problem is solved by Hessian Eigenmaps for flat manifolds.

4.2 The Torus

In the second experiment we tested the method on a data set consisting of 2000 points from a torus embedded in 3-D. The results in figure 4 illustrate how the method cuts the coordinate chart at the cut locus of the point p. This particular behavior of "cutting up" the manifold allows us to save one dimension in this particular example. There is no embedding of the torus into $\mathbb{R}^2$. Any standard method for dimension reduction, e.g. LLE, Laplacian Eigenmaps or ISOMAP, would embed this manifold into $\mathbb{R}^3$ at best. However, the automatic introduction of a cut by the LOGMAP method makes it possible to make a one-to-one mapping of this manifold to $\mathbb{R}^2$.

4.3 The Klein Bottle

The third experiment finally, shown in figure 5, tests the method on truly high-dimensional data. The data set consists of 21×21 pixel image patches. Each of the 2-D image patches were rendered as a 1-D sine wave pattern with a specific phase and orientation. A small amount of normal distributed white noise was also added to the images. The resulting data set consisted of 900 data points, in a 441-dimensional space, representing image patches sampled uniformly from all possible values of phase and orientation . It is natural to assume that the intrinsic dimensionality of this data set is 2, since the variables phase and orientation adds one degree of freedom each.

We observed slightly different shapes of the cut locus, i.e. the border of the resulting map, depending on the choice of base point p. This was somewhat unexpected, but it has a logical explanation. Even though the data set seems to be highly symmetric in terms of orientation and phase, the square shape of the image patches themselves will break the otherwise expected rotation invariance and introduce variations in curvature on the manifold.

The mapping of image patches to $\mathbb{R}^2$ is visualized by simply using the image patches as glyphs, placed at various locations in the plane. By carefully identifying the edges of the cut locus, we manually obtain an interpretation of the mapping shown in the top left of figure 5. This directed labeled graph reveals that the topology of this particular image manifold is actually the well known Klein bottle [12]. Similar conclusions for the topology of local descriptions of phase and orientation has previously been described in [10, 9], where the topology of Gabor filters is derived from theoretical investigations. Our investigation is on the contrary experimental, and to the best of our knowledge it is a new example of how manifold learning can be used to experimentally infer the topology of a data set.

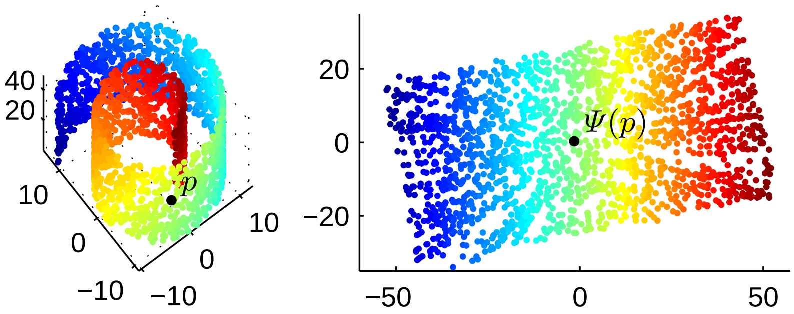

Fig. 3. A set of 2000 points from the "Swiss roll" example [11]. Colors correspond to the first Riemannian normal coordinate derived from the method. **Left:** The original point cloud embedded in 3-D. **Right:** Points mapped to 2-D Riemannian normal coordinates.

5 Discussion

The presented LOGMAP method is rather different from many other methods for manifold learning and dimension reduction, both in terms of the output and in terms of algorithmic building blocks.

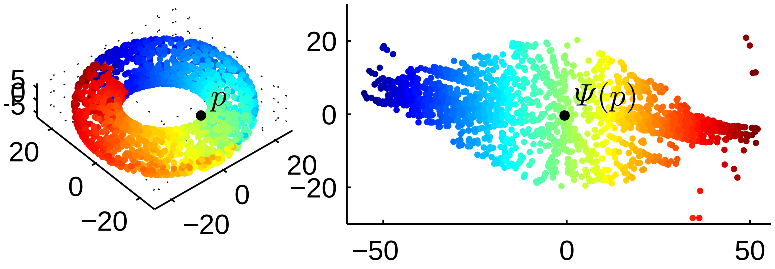

Fig. 4. A set of 2000 points from a torus embedded in 3-D. Colors correspond to the first Riemannian normal coordinate derived from the method. **Left:** The original point cloud embedded in 3-D. Notice the discontinuity (red-blue) in the coordinate map, revealing a part of the "cut locus" of p. **Right:** Points mapped to 2-D Riemannian normal coordinates. Because the metric of a torus embedded in 3-D is not flat, the manifold is not mapped to a perfect rectangle. Some outliers are present, due to incorrect estimation of the gradient for points near the cut locus.

The possibility of a cut, a discontinuity in the mapping at the so called cut locus, could be seen not only as a problem but also as a feature of the method. This allows for instance the torus and the Klein bottle to visualized using a two-dimensional plot. Other methods, such as [11, 7, 8, 1, 2], tries to find a continuous embedding of the manifold, and for that at least 4 dimensions are needed for the Klein bottle and 3 for the torus. (The top middle illustration in figure 5 is actually an example of an immersion and not an embedding of the Klein bottle in 3-D, meaning roughly that it intersects itself at some points.)

The use of other criteria for assigning a global coordinate system to a manifold could also be considered, for instance conformal mappings of 2-D manifolds. In almost every case when mapping a manifold to a low-dimensional space, some kind of distortion is introduced while some features of the original manifold will be preserved. For most manifolds, Riemannian normal coordinates create a very distorted mapping far away from the base point p, in some cases they even introduce a cut. However, they also preserve all geodesic distances and angles from p to other points on the manifold, which makes this mapping quite intuitive and particularly useful for the purpose of navigating inside a manifold. At least this is true in the vicinity of the base point p.

The LOGMAP method is built up by two major algorithmic building blocks:

1. Approximation of distances on a manifold given a set of sampled data points.
2. Calculation of gradients on manifolds, from a set of function values defined at the sampled data points.

For the first building block we have here used Dijkstra's method, mainly inspired by the ISOMAP implementation. This method has obvious problems to truthfully approximate distances, because distances are measured along zigzag trajectories in a graph. One way to make LOGMAP more accurate is therefore to switch to a more accurate method based on higher order approximations of the manifold.

The second building block, which is about calculating gradients, could also be improved a lot compared to the current implementation. Measuring gradients for smooth functions is not a problem, but for points close to the cut locus the distance function will introduce a discontinuity which makes the problem quite delicate. The difficulty of gradient estimation manifests itself by producing spurious points in the mapping, most easily seen in the torus and the Klein bottle examples, close to the cut locus.

In this paper we have chosen examples of manifolds with low intrinsic dimensionality, mainly to illustrate the method, but in principle the method works for manifolds of higher dimensionality too. In the examples we have also used only little or no noise. While this can be seen as very optimistic assumptions about the data, we would like to stress the fact that the LOGMAP method does not try to explicitly deal with noise. In order to handle noise efficiently, it should either be removed prior to the use of LOGMAP or handled by more robust versions of the distance and gradient estimation steps within the LOGMAP framework. This is clearly an important area of future research.

Regarding the efficiency or speed properties of the LOGMAP method, it is important to mention that it, in contrast to many other methods for manifold learning, does not involve the solution of any large eigenvalue problem or any other large scale iterative minimization procedure. Instead it relies totally on the ability to fast approximate distances on the manifold and calculate gradients. A key observation is also that distances $d(x, y)$ are only calculated for pairs $(x, y) \in X \times B(p)$. This is far less demanding than calculating the distance for all pairs $(x, y) \in X \times X$, which is done in for instance ISOMAP.

In summary, we have introduced a novel method for manifold learning with interesting mathematical and computational properties. We have also provided an example of how manifold learning can assist in identifying a rather non-trivial manifold, in this case a Klein bottle, from a high-dimensional data set. We believe this to be of general interest to people within the fields of manifold learning and medical image analysis, to for instance develop better tools for shape analysis, and to inspire the future development of manifold learning and manifold-valued signal processing in general.

6 Acknowledgements

We gratefully acknowledge the support of the Manifold Valued Signal Processing project by the Swedish Research Council (Vetenskapsrådet, grant 2004-4721) and the SIMILAR network of excellence (http://www.similar.cc), the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609).

References

1. Belkin, M., Niyogi., P: Laplacian eigenmaps and spectral techniques for embedding and clustering, in T. G. Diettrich, S. Becker, and Z. Ghahramani (eds:) Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, 2002.
2. Donoho, D. L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data PNAS, vol. 100, no. 10, pp. 5591–5596, May 13, 2003.
3. Fischler, M. A., Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

4. Fletcher, P.T., Joshi, S., Lu, C., Pizer, S.: Gaussian Distributions on Lie Groups and Their Application to Statistical Shape Analysis, In proc. of Information Processing in Medical Imaging (IPMI), pages 450–462, 2003.
5. Kohonen, T.: Self-organized formation of topologically correct feature maps, Biological Cybernetics, vol. 43, pp. 59–69, 1982.
6. Lee, J., Shin, S. Y.: General construction of time-domain filters for orientation data, IEEE Transactions on Visualization and Computer Graphics, vol. 8, no. 2, pp. 119–128, 2002.
7. Roweis, S.,, Saul, L.: Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500): 2323–2326, 22 December 2000.
8. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5): 1299–1319, 1998.
9. Swindale, N.V.: Visual cortex: Looking into a Klein bottle, Current Biology 6(7), pp. 776–779, 1996.
10. Tanaka, S.: Topological analysis of point singularities in stimulus preference maps of the primary visual cortex, Proc. R. Soc. Lond. B, 261: 81–88, 1995.
11. Tenenbaum, J. B., de Silva, V., Langford, J. C.: A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500): 2319–2323, 22 December 2000.
12. Weisstein, E. W.: ”Klein Bottle.” From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/KleinBottle.html

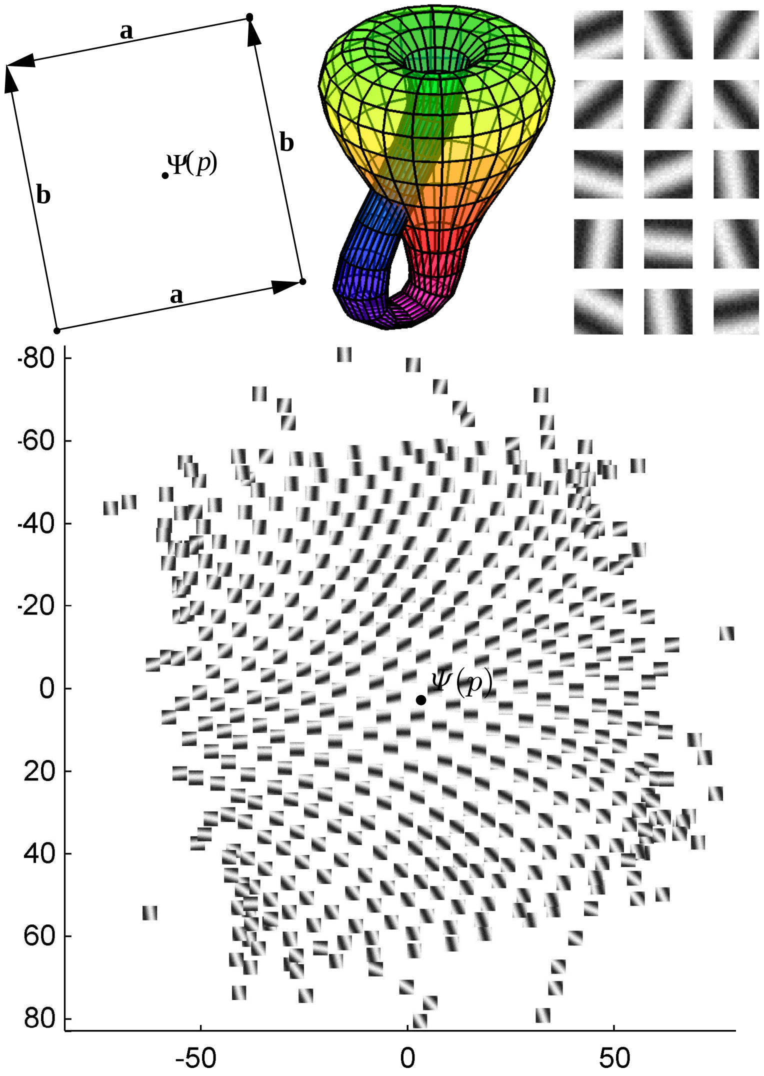

Fig. 5. To test the proposed method on a high-dimensional data set, a set of 900 image patches, each being of 21×21 pixels with a characteristic orientation/phase, were generated and mapped to Riemannian normal coordinates. This experiment reveals the Klein bottle-structure of local orientation/phase in 2-D image patches! **Top left:** An idealized Klein bottle aligned to the mapping below. Edges correspond to the cut locus of p and should be identified according to the arrows. **Top middle:** An immersion of the Klein bottle in 3-D. **Top right:** 15 random examples of image patches used in the experiment. **Bottom:** The mapping of image patches to Riemannian normal coordinates using the proposed method.