

# NON-LINEAR DIMENSION REDUCTION WITH TANGENT BUNDLE APPROXIMATION

J. G. Silva , J. S. Marques, J. M. Lemos

ISEL/ISR, R. Conselheiro Emidio Navarro, 1950-062 Lisboa, Portugal  
IST/ISR, Av. Rovisco Pais, 1949-001 Lisboa, Portugal  
INESC-ID/IST, R. Alves Redol, 9, 1000-029 Lisboa, Portugal

## ABSTRACT

The problem of non-linear dimension reduction is relevant to many different areas of knowledge. While the linear case can be solved by variations of PCA, the non-linear case is more complex. Recent advances incorporate geometrical information by estimating a manifold that approximates the data. The work presented here follows that trend and tackles some limitations of existing approaches: arbitrary topology and curvature of the manifold, unknown intrinsic dimension and, for mixture models, unknown number of mixture components.

An algorithm, designated TBA, is presented that addresses the enumerated difficulties and is faster than existing methods, in datasets of a few thousand points. The motivation behind TBA is to perform motion tracking in video sequences, but the algorithm can be applied in a wide class of problems.

The paper starts with a brief review of related work and then describes the TBA approach in detail. The algorithm is then subjected to comparative evaluation.

## 1. INTRODUCTION

The problem of non-linear dimension reduction is closely related to that of *feature extraction* in pattern recognition literature, and it is important for a number of reasons. If solved, it can yield a concise and meaningful representation of a given set of data, at least in the case where the data belong to an intrinsically low-dimensional manifold embedded in high-dimensional observation space. This allows data compression, complexity reduction, and increased robustness to outliers, for example.

A situation where such a problem arises is motion analysis from video sequences. In the worst-case scenario, if no contours or other features are available, each image in a sequence is an observation, a point in a space which has a dimension equal to the number of pixels - typically tens of thousands. When faced with this type of problem, it is common in machine learning to perform Principal Component Analysis (PCA), thereby extracting a *linear* subspace spanned by the directions of maximum variance of the data.

Another common method, particularly when distances between points are available, rather than their coordinates, is multidimensional scaling (MDS) [1]. The end result (depending on the metric) is typically a linear approximation, like in PCA. However, linear methods not only tend to overestimate the intrinsic dimension of non-linear manifolds, but also often fail to deliver a meaningful representation of the data, particularly when curvature is high.

This work was partially supported by FCT POCTI, under project 37844.

## 1.1. Related Work

Recent advances in non-linear methods such as Generative Topographic Mapping (GTM), [2], Locally Linear Embedding (LLE), [3] and ISOMAP [4], all attempt to learn an intrinsically low-dimensional manifold embedded in observation space.

In GTM, a probabilistic model is also sought, which is not the case with LLE and ISOMAP. Also, neither LLE nor ISOMAP are mixture models, so while GTM requires an *a priori* fixed number of mixture components, LLE and ISOMAP do not. On the other hand, GTM is heavily dependent on initialization and fails frequently when the manifold is not close to linear.

ISOMAP is a *global* method, i. e., it attempts to preserve distances between faraway points as well as nearby ones. It is based on applying MDS to *geodesic* distances, which are approximated by a graph that connects only neighboring points, thus avoiding unwanted short-circuits between nearby folds of the manifold. Although ISOMAP has stronger theoretical guarantees of convergence to the true manifold than LLE, both methods are essentially suited to locally flat manifolds. A newer version, called C-ISOMAP (C for Conformal) [5] is able to deal with curvature. It does not, however, deal with non-trivial topologies, such as that of a sphere or a torus. Also, it relies on routing algorithms, such as Dijkstra, to find the geodesic graph. The complexity of such algorithms is quadratic in the number of points, which renders ISOMAP's computing requirements prohibitive for much more than two or three thousand points. Although this problem can be circumvented by a variation called L-ISOMAP (L for Landmark) [5], the issue remains of which points to use as landmarks.

## 1.2. Present Contribution

Tangent Bundle Approximation (TBA), by contrast, is a local method. It finds multiple representations of the manifold, valid in different regions, and does not require computation of geodesic distances, which makes it considerably faster than ISOMAP. The number of local models does not need to be known in advance, and curvature is well tolerated, as are complex topologies.

The present paper builds on previous work, namely [6] and [7], where the TBA algorithm was introduced. It should be noted that TBA is meant for motion tracking applications, and therefore includes dynamic learning procedures, which are not addressed in this paper. The focus here is on the two following aspects: i) The manifold learning method; ii) A comparative evaluation of TBA with ISOMAP.

An overview of TBA is given in the next section, followed by a detailed explanation. The later sections cover some experimental results, followed by concluding remarks.

## 2. OVERVIEW

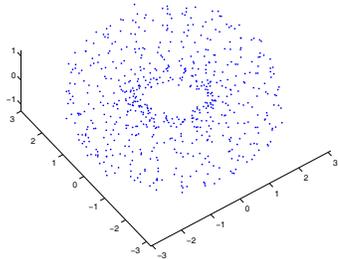
The model for a manifold  $\mathcal{M}$  comprises a set of diffeomorphisms  $g_i : U_i \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ , where  $n$  is the manifold dimension,  $m$  is the dimension of the embedding space and the  $U_i$  are open sets. The  $g_i$ , also called charts, have overlapping images  $g_i(U_i)$ , also called patches, which cover the manifold. As diffeomorphisms, the charts admit inverses  $g_i^{-1}$ .

As the name indicates, TBA approximates the tangent bundle of the manifold. The tangent bundle of the manifold is the set of all tangent vectors at all manifold points, together with a mapping from the manifold points to the tangent vectors [8]. Infinitely many such points and vectors exist, so a discrete and parsimonious approximation is required.

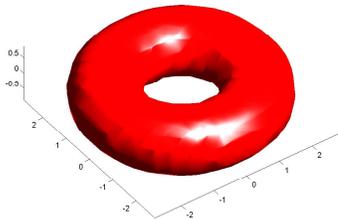
The approximation is done through a soft partitioning based on the *maximum principal angle* (defined in the next section) between neighboring tangent subspaces. Tangent subspaces are found using local PCA - an approach common in similar problems, e. g. surface reconstruction in 3D [9]. The novelty, however, consists of using the maximum principal angle as a similarity measure for the partitioning procedure. This allows charting the manifold in arbitrary topologies, by using the orthogonal basis of the tangent subspaces as coordinate systems and then solving a number of regression problems equal to the number of tangent subspaces.

The TBA algorithm is valid for any number of dimensions, both of the manifold and the embedding space. The only assumptions about the manifold are that it must be smooth and compact, in order to allow covering by a finite number of charts.

The starting point is a training set of scattered, noisy points  $\mathbf{y} \in \mathbb{R}^m$  in observation space, such as the one in Figure 1 (a).



(a) Noisy observations (625 points)



(b) Estimated manifold ( $\epsilon = 0.7$ ,  $\tau = 1$  radian)

**Fig. 1.** Example: a torus, estimated by TBA from observations.

The algorithm follows two main steps: i) Partitioning the data and finding local coordinate systems; ii) Estimating charts. Each of these steps is addressed in their respective sections.

## 3. PARTITIONING

When charting a manifold with arbitrary topology, more than one patch may be required to avoid metric distortion - the so-called cartographer's dilemma. Note that in some situations, such as a torus, which has zero total curvature, one single patch would be theoretically enough. The same is not true, however, in the case of a sphere (which requires two patches), nor in more complex cases.

### 3.1. Intrinsic Dimension

As a first step, the smallest eigenvectors of the local covariance matrix are found for a neighborhood around each data point. These smallest eigenvectors define a normal subspace to the manifold, while the largest eigenvectors define a tangent subspace. This is, essentially, local PCA.

In this step, an important decision must be made: what is  $n$ , that is, the intrinsic dimension of  $\mathcal{M}$ ? TBA addresses this by automatically finding the "knee" of the eigenvalue plots for *all* tangent subspaces at all points. The eigenvalue immediately before the greatest drop in value should correspond to the true intrinsic dimension. The median for all eigenvalue plots is therefore used as the estimate of  $n$ .

### 3.2. Principal angles between nearby normal subspaces

To make chart estimation easier, it is required for simple projection to give a one-to-one mapping between the hyperplane and the corresponding manifold region. This can be ensured by not allowing the maximum principal angle between normal subspaces to vary more than a set threshold  $\tau$ . The exact value of  $\tau$  is not critical, as long as it is below  $\frac{\pi}{2}$ . Therefore, the normal subspaces and their principal angles must be calculated.

The  $q$  principal angles between subspaces spanned by the columns of matrices  $A$  and  $B$  are defined, as in [10], by

$$\cos \theta_k = \frac{|\mathbf{u}_k A^T B \mathbf{v}_k|}{\|A \mathbf{u}_k\| \|B \mathbf{v}_k\|} \quad (1)$$

with  $k = 1, \dots, q = \dim(A) = \dim(B)$  and, for  $k > 1$ , subject to recursively defined constraints for  $\mathbf{u}$  and  $\mathbf{v}$ , the columns of  $A$  and  $B$ :

$$\mathbf{u}_i^T A^T A \mathbf{v} = \mathbf{v}_i^T B^T B \mathbf{v}_k \quad (2)$$

for  $i = 1, 2, \dots, k - 1$ .

An efficient implementation can be found in MATLAB's **subspace** command. The maximum of  $\theta_k$  is used as TBA's intra-patch similarity measure.

Normal subspaces are computed by visiting all data points and, for each one, finding the  $m - n$  smallest eigenvectors of the covariance in a neighborhood of radius  $\epsilon$ . Naturally, this is an unknown scale parameter which strongly influences the algorithm. This is the case with LLE and ISOMAP as well.

### 3.3. Region Growing

Next, patches are found by region growing. Each patch grows by appending all neighboring points where the normal subspace does not deviate, in maximum principal angle, more than a set threshold from the normal subspace at the initial seed. Any specific data point may belong to more than one patch. The following pseudo-code illustrates the procedure:

```

while M not covered
  P = new patch
  y0 = choose a new seed from data points
      not in any patch
  {n0}= normal subspace basis at y0
  while NOT all points visited
    y1 = choose nearest neighbor
    n1 = normal subspace basis at y1
    if principal_angle( n0, n1 ) < tau
      AND distance( y0, y1 ) < epsilon
      append y1 to P
    end if
  end while
end while

```

The final result is a covering of  $\mathcal{M}$  by a finite number,  $p$ , of overlapping patches. Within each patch, the normals don't deviate more than  $\tau$ , and the distance test ensures that each patch is a connected set. Subsequently using patch-wide PCA (with all patch members - not the same as local PCA above) to find the best fitting hyperplane provides a number of local coordinate systems, valid in different manifold regions. The best hyperplane, in a least squares sense, is spanned by the  $n$  largest eigenvectors returned by patch-wide PCA. Each patch is thus associated to an hyperplane, and the collection of hyperplanes approximates the tangent bundle.

An important note is that TBA does not guarantee that the number of patches is minimal - in fact, the followed approach leads to an overestimation of the number of patches needed to cover a manifold.

#### 4. CHART ESTIMATION

Charts are then estimated, using the coordinate systems found above. Since the subspace angle was only allowed to change up to a specified limit, it is easy to ensure that the charts are bijective. The estimation procedure is designed to ensure that the charts are also differentiable, so they are diffeomorphisms, as intended. It is important to note that, since there are no folds in any patch (thanks to the angular restriction), the regression problems are significantly simplified.

There are many alternatives for non-linear function approximation that meet the problem's requirements. The results presented here were obtained using thin-plate splines [11], but any other function approximator can be used, as long as it doesn't require regularly gridded data - RBF or MLP neural networks are good possibilities as well.

From the previously obtained partition of  $\mathcal{M}$  in  $p$  patches, it is possible to find  $p$  charts  $g_i(\mathbf{x})$ . With points  $\mathbf{y} = [y^1 \dots y^m]^T$  belonging to a given patch  $i$ , and having previously performed PCA, a matrix  $V_i$  of eigenvectors and a mean vector  $\mu_i$  are available. Projecting  $\mathbf{y}$  on the hyperplane associated with patch  $i$  can be done according to

$$\tilde{\mathbf{x}} = V_i^T (\mathbf{y} - \mu_i) \quad (3)$$

$$\mathbf{x} = [\tilde{x}_1 \dots \tilde{x}_n]^T \quad (4)$$

As for  $g_i$ , which is the inverse mapping of (2), it follows the expression

$$g_i(\mathbf{x}) = V_i \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \mu_i \quad (5)$$

The remaining  $m - n$  components of  $\tilde{\mathbf{x}}$ , could be set to zero, which would yield a piecewise linear approximation of  $\mathcal{M}$ . Instead, however, they are kept, which increases the approximation precision and preserves curvature. In local coordinates, the manifold parametrization is

$$\mathbf{x} \rightarrow [\mathbf{x} \quad \tilde{g}(x)]^T \quad (6)$$

where  $\tilde{g}$  denotes the approximating thin-plate spline.

## 5. RESULTS

To illustrate both the manifold learning and the dynamic learning results in different circumstances, two synthetic examples are presented, followed by a video sequence example with real data.

### 5.1. Synthetic Examples

In the first synthetic example, a 2-torus embedded in  $\mathbb{R}^3$ , as in Figure 1 (a), it is shown that the TBA algorithm can reconstruct non-trivial shapes from noisy data - 625 points in a few seconds, in this case. The estimated manifold is represented in Figure 1 (b). The torus is, as seen, accurately reconstructed. Neither ISOMAP/C-ISOMAP nor LLE can deal with this example, because of its topology. The parameters used are  $\epsilon = 0.7$  and  $\tau = 1$  radian ( $< \pi/2$ ).

The second example is a swiss-roll dataset similar to the one used to demonstrate ISOMAP in [4], but now with 2601 points on a regular grid instead of 1000 randomly positioned points. For this example, the results of both ISOMAP and TBA are presented. The code for ISOMAP is the one available on <http://isomap.stanford.edu>, and the neighbourhood parameter was  $k = 8$  nearest neighbours. For TBA,  $\epsilon = 0.2$  and  $\tau = 1$  radian were the values used.

In Figure 2, the global embedding returned by ISOMAP is shown, in the form of an adjacency graph, since ISOMAP does not include a method for continuous interpolation at points other than those in the training set. For TBA, the returned patches are computed for points *not* included in the training set and represented in observation space, overlaid on the original points and the 11 returned patch centers (red circles). Both ISOMAP and TBA accurately find the true manifold, as seen in the figure. However, ISOMAP takes 89 minutes, while TBA takes 4 minutes. Both tests were made using a Pentium 4 laptop at 1.8 GHz with 512 MB of RAM.

### 5.2. Video Sequence

The training set for this example consists of  $M=194$  cropped  $71 \times 71$  grayscale images from the same person (a TV presenter), and it is intended to learn the manifold of valid face configurations. The mean image is subtracted so that the training set has zero mean. A more detailed description of this example can be found in [7].

Briefly, using the eigenface method [12] it is possible to express all images as linear combinations of the first  $M'$  eigenvectors (also called eigenfaces) without too much loss of information. In this case,  $M' = 15$  is a reasonable value.

Therefore, a linear transformation is applied to the data and only the top  $M'$  rows of the transformed data  $W$  are kept, reducing

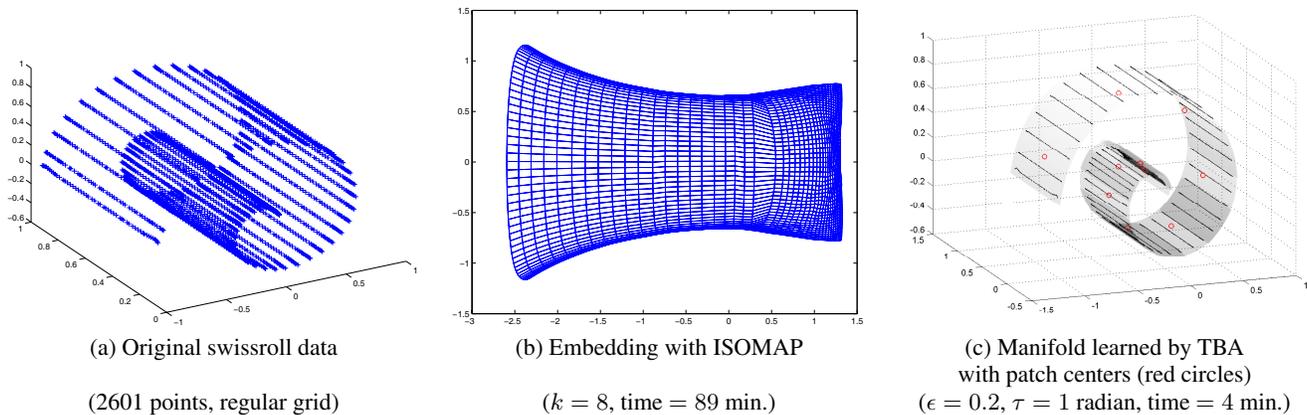


Fig. 2. Example: swiss roll data, with ISOMAP and TBA.



Fig. 3. Example: images from a video sequence.

the problem dimensionality from  $71 \times 71 = 5041$  to 15. The resulting  $15 \times 194$  matrix contains the new training vectors for TBA.

The algorithm returns two patches with intrinsic dimension 3, although there is no clear "knee" for the eigenvalues in this case. Even using only 3 degrees of freedom instead of 15, the reconstructed sequence is quite similar to the original, as seen in Figure 3. An interesting development is that interpolation between faces from the training set is possible, thanks to the continuous charts, and allows synthesis of new, valid face images.

## 6. CONCLUSIONS

This paper presents an approach for dimension reduction and manifold learning, based on the Tangent Bundle Approximation (TBA) algorithm. Results are presented for synthetic examples and a face example with real data. A comparative evaluation with ISOMAP shows that TBA is at least an order of magnitude faster, and can be used in a wider class of problems. TBA also returns continuous charts that can be used for interpolation. Although the same could be done in ISOMAP, taking advantage of the global discrete mapping it provides, highly folded manifolds would make such interpolation difficult - a problem TBA avoids by restricting variation of the maximum principal angle of the normal subspace within each patch.

Issues for future work include reducing the number of patches, for a more concise model, as well as a way to compute the scale parameter  $\epsilon$ .

## 7. REFERENCES

- [1] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman & Hall, London, 1994.
- [2] C. Bishop, M. Svensen, and C. Williams, "Gtm: The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215, 1998.
- [3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
- [4] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
- [5] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *NIPS*, vol. 15, 2002.
- [6] J. Silva, J. Marques, and J. M. Lemos, "A geometric approach to motion tracking in manifolds," *IFAC Symposium on System Identification*, 2003.
- [7] J. Silva, J. Marques, and J. M. Lemos, "Motion tracking in manifolds with tangent bundle approximation," *IFAC Symposium on Nonlinear Control Systems*, 2004.
- [8] B. O'Neill, *Elementary Differential Geometry*, Academic Press, 1997.
- [9] H. Hoppe, *Surface reconstruction from unorganized points*, Ph.D. thesis, University of Washington, 1994.
- [10] A. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematical Computation*, vol. 27, 1973.
- [11] T. Duchamp and W. Stuetzle, "Spline smoothing on surfaces," *Journal of Computational and Graphical Statistics*, vol. 2, pp. 354–381, 2003.
- [12] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.