Intrinsic Dimensionality Estimation with Optimally Topology Preserving Maps

J. Bruske, G. Sommer
Computer Science Institute, Christian-Albrechts University Kiel
Preusserstr. 1-9, 24105 Kiel, Germany
email:jbr@informatik.uni-kiel.de

February 1997
Technical Report Nr. 9703*

Abstract

A new method for analyzing the *intrinsic dimensionality* (ID) of low dimensional manifolds in high dimensional feature spaces is presented. The basic idea is to first extract a low-dimensional representation that captures the *intrinsic topological structure* of the input data and then to analyze this representation, i.e. estimate the intrinsic dimensionality. More specifically, the representation we extract is an *optimally topology preserving feature map* (OTPM) which is an undirected parametrized graph with a pointer in the input space associated with each node. Estimation of the intrinsic dimensionality is based on *local PCA* of the pointers of the nodes in the OTPM and their direct neighbors. The method has a number of important advantages compared with previous approaches: First, it can be shown to have only *linear time complexity* w.r.t. the dimensionality of the input space, in contrast to conventional PCA based approaches which have cubic complexity and hence become computational impracticable for high dimensional input spaces. Second, it is *less sensitive to noise* than former approaches, and, finally, the extracted representation can be directly used for further data processing tasks including auto-association and classification.

Experiments include ID estimation of synthetic data for illustration as well as ID estimation of a sequence of full scale images.

*This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible

1 Introduction

The intrinsic, or topological, dimensionality of N patterns in an n-dimensional space refers to the minimum number of "free" parameters needed to generate the patterns[1], [3]. It essentially determines whether the n-dimensional patterns can be described adequately in a subspace (submanifold) of dimensionality $m < n$. Knowledge of the intrinsic dimensionaliy is important in order to determine the number of features necessary to represent the data, to decide whether a reasonable 2d or 3d representation exists or to estimate the effectiveness of algorithms depending on the intrinsic dimensionaliy, as e.g. methods for constructing classifiers or training neural networks. It can be greatly helpful in problems like pattern recognition, industrial or medical diagnosis and data compression [4].

Adopting the classification in [3], there are two primary approaches for estimating the intrinsic dimensionality. The first one is the *global approach* in which the swarm of patterns is unfolded or flattened in the d-dimensional space. Benett's algorithm [5] and its successors as well as variants of MDSCAL [6] for intrinsic dimensionality estimation belong to this category. The second approach is a *local* one and tries to estimate the intrinsic dimensionality directly from information in the neighborhood of patterns without generating configurations of points or projecting the pattterns to a lower dimensional space. Pettis' [7], Fukunaga and Olsen's [8] as well as Trunk's [9] and Verveer and Duin's method [10] belong to this category.

Our approach belongs to the second category as well and is based on local principal component analysis (PCA) using a number of evenly distributed pointers in the manifold. The denser these pointers the more accurate the local estimate provided by the PCA, i.e. the number of eigenvalues which approximates the intrinsic dimensionality at this point. However, given the covariance matrix of some distribution in an n-dimensional vector space, PCA of this covariance matrix takes time $O(n^3)$. Hence the computational cost

[1]It has long been noticed that this 19th century notion of dimensionality is unprecise and fraught with problems, see e.g. [1] for a short review. Yet there exists a precise definition of the *topological dimensionality*, given by Blouwer in 1913, [2]. It is this type of dimensionality we try to estimate, as opposed to the *fractal* or *Hausdorff* dimension. In spite of its insufficiencies the intuitive definition of dimensionality as the number of continuous parameters needed to describe a set of points has prevailed throughout the pattern recognition literature. And because it is so intuitive we will stick to it as well.

becomes prohibitive for higher dimensions. This problem is circumvened in the following way: After distributing the pointers in the manifold M we first extract a low dimensional representation of M by constructing an optimally topology preserving map ($OTPM$). In an $OTPM$ two nodes are connected if their associated pointers are neighbored in M. Due to this definition the $OTPM$ does only depend on the intrinsic structure of the manifold and is independent of the dimensionality of the embedding input space. Since the number of neighbors m_i of a node in an OTFM is small for low dimensional submanifolds and because of the independence of n, it will usually be much smaller than n for high dimensional input spaces. Using a well-known trick one can now perform PCA for m points in an n-dimensional space in time $O(m^3)$, independent of n. The calculation of the covariance matrix for these m points takes time $O(m^2 n)$, and hence the time complexity of the procedure grows only linearly with the dimension of the input space.

Real data is always noisy and hence samples stemming from some low dimensional hypersurface will always contain noise orthogonal to the surface. By using a statistical clustering procedure to distribute the pointers prior to construction of the $OTPM$ the pointers nevertheless can be expected to be placed on the surface in spite of the noise. In this situation of a pointer and its topological neighbors all lying on the surface, local PCA of the neighboring points will not detect any variance orthogonal to the surface (except the contribution of curvature). On the contrary, simple PCA of the data distribution in the Voronoi cell of a pointer would always contain the variance of the noise. Hence, besides being impractical, the eigenvalues produced by straight forward PCA are less suited for discrimination between the noise and the surface.

The rest of this paper is organized as follows: In section 2 we will have a closer look at $OTPM$s, the representation underlying our intrinsic dimensionality estimation method, describe a trick for efficient PCA for $m < n$ points, the method we use for analysing the representation and finally comment on the problem of estimating the ID by local PCA in general. We will then state our algorithm more precisely in section 3 including a brief discussion on the issue of vector quantisation. Experimental results are given in section 4, related work is discussed in 5 and we give some closing remarks in 6.

2 Foundations

In this section we want to make the reader familiar with the basic ingredients of our algorithm for ID estimation to be presented in the next section. We fist introduce $OTPM$s, the underlying representation, and then turn to efficient PCA for $m < n$ points, the underlying method used for analyzing the $OTPM$, and finally comment on the problem of estimating the ID by local PCA, the general approach of our algorithm.

2.1 Constructing Optimally Topollogy Preserving Maps

Optimally Topology Preserving Maps ($OTPM$s) are closely related to Martinetz' Perfectly Topology Preserving Maps (PTPMs) [11] and are constructed in just the same way. The only reason to introduce them separatly is that in order to form a PTFM the pointers must be "dense" in the manifold M. Without prior knowledge this assumption cannot be checked, and in practice it will rarely be valid. $OTPM$s emerge if just the construction method for PTFMs is applied without checking for the density condition. Only in favourable cases one will obtain a PTFM (probably without noticing). $OTPM$s are nevertheless optimal in the sense of the topographic function introduced by Villmann in [12]: In order to measure the degree of topology preservation of a graph G with an associated set of pointers S, Villmann effectively constructs the $OTPM$ of S and compares G with the $OTPM$. By construction, the topographic function just indicates the highest (optimal) degree of topology preservation if G is an $OTPM$.

Definition 1 (OTPM) *Let $p(x)$ be a probability distribution on the input space R^n, $M = \{x \in R^n | p(x) \neq 0\}$ a manifold of feature vectors, $T \subseteq M$ a training set of feature vectors and $S = \{c_i \in M | i = 1, \ldots, N\}$ a set of pointers in M.*

We call the undirected graph $G = (V, E)$, $|V| = N$, an optimally topology preserving map of S given the training set T, *$OTPM_T(S)$, if*

$$(i,j) \in E \;\Leftrightarrow\; \exists x \in T\, \forall k \in V \backslash \{i,j\} : \max\{\| c_i - x \|, \| c_j - x \|\} \leq \| c_k - x \|$$

Corolary 1 *If $T = M$ and if S is dense in M then $OTPM_T(S)$ is a PTPM.*

Note that the definition of the $OTPM$ is constructive: For calculating $OTPM_T(S)$ simply pick $x \in T$ according $p_T(x)$, calculate the best and second best matching pointers, c_{bmu} and c_{smu}, and connect bmu with smu. If repeated infinitely often, G will converge to $OTPM_T(S)$ w.p.o.. This procedure is just the essence of Martinetz' Hebbian learning rule.

For use in intrinsic dimensionality estimation and elsewhere, $OTPM_T(S)$ has two important properties. First, it does indeed only depend on the intrinsic dimensionality of T, i.e. is independent of the dimensionality of the input space. Embedding T into some higher dimensional space does not alter the graph. Second, it is invariant against scaling and rigid transformations (translations and rotations). Just by definition it is the representation that optimally reflects the intrinsic (topological) structure of the data.

Since we will compute a PCA of the covariance matrix of all the m_i neighbors of a node $v \in OTPM_T(S)$ and the cost of this computation will be $O(m_i^3)$ (section 2.2) it would be nice to have some estimate of the number of neighbors in $OTPM_T(S)$ as a function of the intrinsic dimensionality d of the structure, the number of pointers c and the density function $p(x)$. While experience shows that for low dimensional submanifolds and a limited number of pointers m_i is relatively small, theoretically little is known. Of course, the number of pointers c is an upper bound on m_i, and in degenerated cases (pointer lie on a circle) this bound can be reached. A lower bound can be derived by looking at the simplest polyhedron in d-dimensional space, the hypertetrahedron. It has $d+1$ corners, hence nodes representing a d-dimensional manifold must have at least d neighbors. For a very large number of pointers Frisone et al. [4] hypothesize that the problem bears some resemblance to the problem of the maximum kissing number in SPP (sphere packing problem). The problem here is to find a packing of d-dimensional spheres of equal size so that the number τ of spheres touching (kissing) each other is maximal [13]. Kiss-SPP has only been solved for $d = 1, 2, 3, 8, 24$ (τ = 2, 6, 12, 240, 196560) and there exist optimal solutions for lattices of spheres for $d = 4, 5, 6, 7$ (τ = 24, 40, 72, 126) [13]. The basic assumptions behind this analogy are that first the pointers have been optimally distributed in the manifold (in the sense of the lowest quantization error), second this optimal distribution is a lattice quantizer and third the problem of finding the best lattice quantizer is dual to finding the lattice with highest kissing number. While there is some evidence that the latter two assumptions hold at least for small d, [13], the basic problem with this estimate is the necessity

of a huge number of pointers (and even huger number of samples in T) and their optimal distribution.

Finally, with respect to the construction time of $OTPM$s Martinetz, [14], has shown that for a uniform density function on average $O(|E|\log(|E|))$ sample presentations are necessary, if the pointer distribution is uniform as well. For highly nonuniform pointer distributions serial time complexity will reach $O(|E|^2)$. Of course, for a finite training set T the $OTPM_T(S)$ can be constructed in time $O(|T|)$, simply by calculating the best and second best matching pointer for each $x \in T$.

2.2 Efficient PCA for fewer points than dimensions

We now want to draw the reader's attention to a basic trick from linear algebra which allows to calculate the PCA of the covariance matrix of a set of points $S = \{c_i \in R^n | i = 1, \ldots, N\}$ in time $O(N^2 n + N^3)$. This trick is useful whenever $N < n$, i.e. there are fewer points than dimensions, a situation characteristic for $OTPM$s of low dimensional submanifolds in high dimensional input spaces and frequently encountered in image analysis.

Let $A^T = [c_1, \ldots, c_N]$. The trick is just to calculate the PCA of $\hat{\Sigma} = \frac{1}{N} AA^T$ instead of a PCA of the original covariance matrix $\Sigma = \frac{1}{N} A^T A$. The eigenvalues of Σ, $\mu_1, \ldots, \mu_N$, are then identical to the eigenvalues of $\nu_1, \ldots, \nu_N$ of $\hat{\Sigma}$ and the eigenvectors of Σ, $u_1, \ldots, u_N$, can be calculated from the eigenvectors $v_1, \ldots, v_N$ of $\hat{\Sigma}$ by setting $u_i = A^T v_i$. This can be simply checked by

$$\hat{\Sigma} v_i = \nu_i v_i \Leftrightarrow AA^T v_i = \nu_i v_i \Leftrightarrow A^T AA^T v_i = \nu_i A^T v_i \Leftrightarrow \Sigma(A^T v_i) = \nu_i A^T v_i$$

Since the PCA of the $N \times N$ matrix $\hat{\Sigma}$ can be calculated in $O(N^3)$, [15], and $\hat{\Sigma} = \frac{1}{N} AA^T$ clearly can be computed in time $O(N^2 n)$, it takes indeed time $O(N^2 n + N^3)$ to calculate the PCA of the covariance matrix of S. A brief summary of fast PCA algorithms can be found in [16].

2.3 On the problem of ID estimation with local PCA

Following an analysis similar to that of [8] and [10] we assume the data points $x \in T$ to be noisy samples of a vector valued function $f : R^r \to R^n$

$$x = f(k) + \eta \tag{1}$$

where $k = [k_1, \ldots, k_r]$ is an r-dimensional parameter vector and η denotes the noise. Using the Taylor expansion of f and neglecting higher order terms [2], f can be approximated by a linear function for small parameter variations Δk around k_0

$$\Delta f = f(k_0 + \Delta k) - f(k_0) \approx \Delta k^T \Psi(k_0) \quad \text{with} \quad [\Psi(k_0)]_{ij} = \frac{\partial f_i(k_0)}{\partial k_j} \tag{2}$$

Both the functional form of f and the number of parameters r are unknown and we are only given the noisy samples. Local PCA of the matrix

$$C = E\{\Delta f \Delta f^T\} = E\{(x - x_0)(x - x_0)^T\} \tag{3}$$

i.e. the "covariance"-matrix obtained for observed samples x of f in the vicinity of $x_0 = f(k_0)$ taken as the "mean", yields the K eigenvalues μ_i and orthonormal eigenvectors u_i of C with

$$Cu_i = \mu_i u_i \quad i = 1, \ldots, K \tag{4}$$

These eigenvectors may serve as an alternative basis for the linear approximation of f and we can write

$$\Delta f = \Delta h^T \Theta \quad \text{with} \quad \Theta = [u_1, \ldots, u_K], \tag{5}$$

where Δk and Ψ are related to Δh and Θ by a linear but unknown transformation.

Since Δf is spanned by r or less linearly independent vectors the number K of eigenvalues should be at most r, i.e. $K \leq r$. However, because the data is noisy and the region for taking the samples is not infinitely small, one will usually obtain up to n eigenvalues. Yet if the region and the noise are small enough and if the linear approximation holds, r or less eigenvalues should dominate, and this is the motivation behind using local PCA for intrinsic dimensionality estimation.

As pointed out in [10], we can imagine the effect of noise to render the r-dimensional surface S defined by f not infinitely thin. In any local region we have r vectors s_i spanning the surface and $n - r$ vectors n_j perpendicular to

[2] In general, there is no guarantee that the first-order term dominates the Taylor series. However, our own as well as the experiments of [8] confirm the workability of this assumption for local ID estimation.

S. Within a small region the linear approximation is only valid if the largest variance in direction perpendicular to S is much smaller than the smallest variance in direction of S, i.e.

$$\frac{\min_i Var(s_i)}{\max_j Var(n_j)} \gg 1. \tag{6}$$

Here, $Var(s_i)$, the intra-surface variance, depends on the size of the local region and $Var(n_j)$ depends on the variance caused by the noise *and* the fact that S cannot be exactly represented as a linear surface. This leads to a basic dilemma for any ID estimation algorithm based on local PCA: If the region is too large, $Var(n_j)$ might be high due to the non-linear nature of S. If, on the other hand, the region is too small, the noise is still there and will eventually dominate $Var(s_i)$. The solution is to search for the region size that gives the best compromise[3].

Closely related to the problem of noise is the problem of having available only a limited set of data. In order to make local PCA approaches work, the data set has to be large enough to represent the non-linearities and to allow for filtering out the noise.

3 Dimensionality Analysis with $OTPM$s

The basic procedure *tpca* for intrinsic dimensionality analysis with $OTPM$s is summarized in figure 1. To find a set S of N pointers which reflects the distribution of T the procedure first employs a clustering algorithm for T whose output are N cluster centers. Then it calculates the graph G as the optimal topology preserving map of S given T. The final step is to perform for each node v_i a principal component analysis of the correlation matrix of the difference vectors $c_j - c_i$ of the pointers c_j associated with the nodes v_j adjacent to v_i in G. The result of this analysis, i.e. eigenvalues and vectors for each node, is the output of the procedure and subjected to further analysis. Provided the complexity of the clustering algorithm is independent of the intrinsic dimensionality d the serial time complexity of *tpca* is $O(n +$

[3]Different to [10] and [8] in our ID estimation procedure noise is largely reduced by the additional clustering stage (see below). Thus for the same local region size we will usually obtain much higher values for the expression in (6) and can better discriminate between the noise and the surface. Yet the basic dilemma remains.

$m(d,T,S)^3$), where $m(d,T,S)$ is the maximum number of direct neighbors of a node in the $OTPM$ as depending on the intrinsic dimensionality, the training set T and the set of pointers S. As already discussed, bounds on $m(d,T,S)$ or even a functional form are hard to derive, yet m stays constant for constant ID, is independent of the input dimension n and experiments confirm that it is indeed small for small IDs.

In the rest of this section we will first comment on the use of clustering algorithms for *tpca* and then extend the procedure to derive our actual ID estimation method.

input training set $T \subseteq M \subseteq R^n$, number of pointers N

procedure tpca(T,N) {
  S = Cluster(T,N)
  $G = OTPM_T(S)$
  for_all_nodes $(v_i \in G)$\{
    $Q_i = \{(c_j - c_i) | c_i, c_j \in S; (v_i, v_j) \in E_G\}$
    output PCA(cor(Q_i))
  }
}

Figure 1: tpca: Topology aided Principal Component Analysis

3.1 Clustering in TPCA

The reason for clustering the data prior to construction of the $OTPM$ and not just drawing N pointers randomly from T is twofold: First the distribution of the pointers should reflect the underlying distribution $p_T(x)$ as accurately as possible and second we would like to eliminate noise on the data. Any vector quantization algorithm which aims at minimizing the (normalized) quantization error

$$J = \frac{1}{n} \sum_{i=1}^{N} \int_{V_i} \| x - c_i \|^2 \, p(x) dx, \tag{7}$$

where V_i denotes the Voronoi cell of c_i, is a good choice since by minimizing the total variance it will preferably place the pointers within the manifold M and filter out orthogonal noise. This holds because as long as criterion (6) is fulfilled placing pointers within the surface and hence reducing the intra-surface variance causes the largest decrease in J. It also produces a distribution of pointers which reflects the probability density. More specifically, for a quantizer minimizing J it holds that (for large numbers of pointers) the density of pointers $P(x)$ is related to the input probability density $p(x)$ via

$$P(x) = p(x)^{\mu} \tag{8}$$

with $\mu = n/(n+2)$ the *magnification factor* [4]. Hence for a uniform probability distribution or high dimensional input spaces minimization of J performs very well in reproducing the underlying density. Probably the most common vector quantization algorithm for minimization of J is the LBG algorithm [18], [19]. Alternatively, the calculation of the centroids s_i can be formulated as a stochastic on-line process, [20]. Closely related to stochastic minimization of J and hence appropriate for use as clustering algorithms as well are the various types of self organizing maps. For the original SOM of Kohonen, [21], no energy function exists. Ritter et. al. [22], however, have shown that under certain assumptions the distribution of pointers can be described by a magnification factor of $\mu = 2/3$. On behalf of his Neural Gas algorithm, Martinetz [14] was able to find an energy function closely related to J and to derive the magnification factor of $\mu = n/(n+2)$, identical to that obtained by minimization of J. The advantage of using the neural type of clustering algorithms is that due to neighborhood cooperation they usually converge much faster than their stochastic counterparts without neighborhood cooperation. A further advantage of stochastic quantizers with neighborhood cooperation is the possibility to actually control the magnification factor as suggested in [23].

3.2 An ID estimation procedure

In order to use *tpca* for ID estimation we must eventually decide how many dominant eigenvalues exist in each local region, i.e. what size an eigenvalue

[4] This follows from the more general result in [17] stating that the reconstruction error $E = \int_M |x - c_i|^p dx$ is minimized by pointer distribution with $\mu = \frac{n}{n+p}$.

as obtained by each local PCA must exceed to indicate an associated intra-surface eigenvector. This amounts to determining a threshold. We adopted the Dα criterion from Fukunaga et. al. [8] which regards an eigenvalue μ_i as significant if

$$\frac{\mu_i}{\max_j \mu_j} > \alpha\%. \tag{9}$$

If no prior knowledge is available, different values of α have to be tested. Otherwise, knowledge of the largest noise component can be used to calculate α.

A second problem is that due to the noise/non-linearity dilemma mentioned in section 2.3 we do not know the optimal local region sizes in advance and, in particular, do not know the optimal number of pointers N as required by procedure *tpca*. Monitoring the development of the local eigenvalues for a growing number of pointers ($N = 1, \ldots$) and searching for characteristic transitions is the most natural way to proceed. In this case, one does not want to cluster all the $N + 1$ pointers from scratch but rather would like to incrementally build on the existing N clusters, i.e. just add one new cluster and modify the existing ones if necessary.

Using the LBG vector quantization algorithm, [18], we start with $N = 1$ and add a new pointer by first searching the cluster with highest intra cluster variance, i.e.

$$\frac{1}{d_i} \sum_{x \in V_i} \| x - c_i \|^2 \geq \frac{1}{d_k} \sum_{x \in V_k} \| x - c_k \|^2 \quad \forall 1 \leq k \neq i \leq N, \tag{10}$$

where d_i denotes the current local ID-estimate at pointer c_i [5]. In this cluster we then search for the training sample x with the highest quantization error, add a new pointer at x, take this configuration of $N + 1$ pointers as the new starting configuration for the LBG algorithm and run *tpca* for the $N + 1th$ round. This procedure of first searching for the worst quantized cluster helps to alleviate problems with outliers which could lead to multiple insertions at the same point if only the worst quantized example was considered.

Finally, if we have reason to believe that the data set has constant intrinsic dimensionality (i.e. has been generated by one function and not by a mixture

[5] We normalize by the local ID-estimate to avoid J being dominated by the quantization error of samples in regions of high intrinsic dimensionality. Of course, if the data set is known to have constant ID this normalization is not necessary.

of functions) our estimate of the intrinsic dimensionality will be the average of all local ID estimates together with its standard deviation. The ID estimate and its standard deviation is then plotted versus the number of pointers N, with different plots resulting from different choices of α. In the next section we will demonstrate that these plots actually do give very fine and characteristic hints on the ID of the data set. Our estimation procedure is interactive because the user has to choose a set of thresholds α and the final decision on the ID depends on his inspection of the ID plots. Yet for reasons already indicated and further illustrated in the next section, without prior knowledge a fully automated procedure based on local PCA which outputs the ID estimate given the data set does not make sense.

4 Experimental Results

In this section we investigate the ID estimation procedure on an experimental basis and also demonstrate it workability for high dimensional real world image data. In the first experiment we apply the procedure to a mixture of noisy data sets of different intrinsic structure and dimensionality. In a second experiment with data stemming from a rectangular surface we will have a closer look at the influence of noise. Further experiments deal with ID estimation of noisy and noise-free surfaces of hyperspheres and Lissajous figures in different dimensions. With respect to ID estimation of high dimensional image data we analyze two image sequences obtained by letting a robot arm turn a) a symmetrical grey ramp and b) a bottle of beer in front of a camera.

4.1 First experiment

Our first experiment is to give a first impression of the characteristics of our ID estimation procedure by applying it to a mixture of noisy data sets of different intrinsic structure and dimensionality. The 3d data set, as illustrated in figure 2, consists of 5000 random dots within a circle, a line and a square in the xy-plane with uniform noise[6] in the z-direction. The circle has a diameter of 6 , the line a length of 6 and the square an area of 6×6 units.

[6]The particular distribution of the noise, e.g. Gaussian or uniform, does only play a minor role because it is averaged out by the clustering procedure. Important is its variance.

The noise has an amplitude of ± 0.5 units (and hence variance of $1/12$). The data density is approximately uniform over the data set.

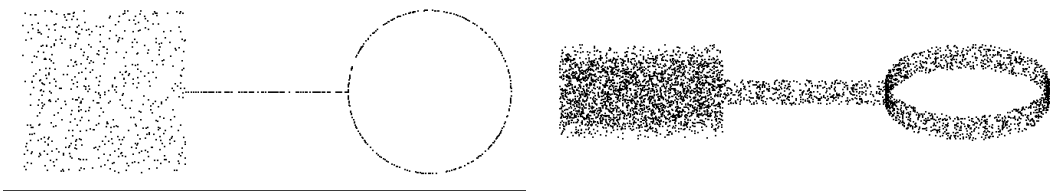

Figure 2: Two views of the Square-Line-Circle data set. Left: View on the xy-plane, Right: Rotation of 60° around x-axis

Figure 3 shows the ID estimation procedure in progress for a growing number of pointers on the D10 level. From top to bottom, left to right with 5, 10, 20, 35, 45, and 70 nodes in the $OTPM$. Dark circles indicate a local ID estimate of one, medium dark circles an estimate of two and light circles of three (D10 criterion). For only five nodes the $OTPM$ indicates a one dimensional connection structure for the circle and the line and a two dimensional one for the square, identical to the ID estimates (by local PCA of the $OTPM$). For 10 nodes the $OTPM$ has already grasped the intrinsic structure of the data set. For 20 nodes we also get the correct local ID estimates for the line-data and the square but the ID estimate of the circle data is still two instead of one. This is due to the curvature (non-linearity) of the circle. From 35 to 45 nodes even the true ID of the circle is revealed because the number of pointers has now become large enough for a linear approximation of the circle on the D10 level. For even higher numbers of pointers the distribution of pointers as obtained by the LBG algorithm will eventually approximate the noise, i.e. leave the surface. From now on (see figure 3 for 70 nodes) the ID will be overestimated.

We want to remark that the mean squared quantization error

$$mse = \frac{1}{|T|} \sum_{i=1}^{N} \sum_{x \in V_i} \| x - c_i \|^2 \tag{11}$$

for e.g. $N = 45$ nodes is 0.29 which is only about three times the variance of the noise. Subtracting the noise variance, only two times the noise variance remains for the average local intra-surface variance. Clearly, a simple local PCA approach as e.g. that of Fukunaga et al. (taking the unfiltered data as

input to the local PCA) would not yield the correct local ID estimates on a D10 level for that local region size but would detect the noise variance as a second or third most significant eigenvalue on any level. This is what we refer to as the increased robustness against noise and the increased discrimination ability of our procedure.

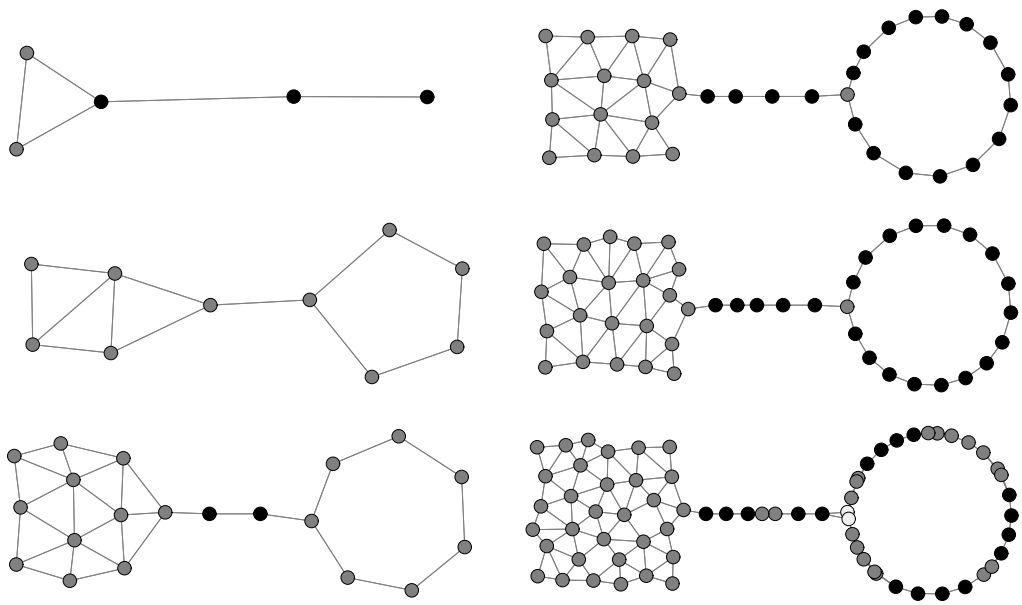

Figure 3: Local ID estimation for the Square-Line-Circle data set for a growing number of pointers (nodes in the $OTPM$) on the D10 level. From top to bottom, left to right: $5, 10, 20, 35, 45, 70$ nodes. Dark circles indicate a local ID estimate of one, medium dark circles an estimate of two and light circles of three dimensions.

4.2 Second Experiment

We now want to take a closer look at how the LBG vector quantization stage distributes the pointers in the manifold and the ID estimation procedure copes with noise. As a data set we choose 5000 noisy data originating from a rectangular surface of 18×3 units in the xy plane. The amplitude of the uniform noise is ± 0.5 in z-direction (variance of $1/12$). The data density is uniform over the data set. The data set is illustrated in figure 4.

Figure 4: Two views of the rectangular data set. Left: View on the data set in xy-plane, Right: Side view on the noise (Rotation of 90° around x-axis)

Figure 5 shows the local ID estimation for the rectangular data set for a growing number of nodes in the $OTPM$ on the D01 (left) and D10 level (right). From top to bottom: 4, 10, 20, 40, 60, 70 nodes. Again, dark circles indicate a local ID estimate of one, medium dark circles an estimate of two and light circles of three.

The figures nicely illustrate how the incremental LBG clustering stage minimizes the quantization error by placing the pointers along the principal axis of the noisy surface in order of decreasing variance along the axis. The first four nodes are placed along the first principal axis and the $OTPM$ as well as the ID estimates indicate a one dimensional line. For 10 and 20 nodes we see how the pointers are also placed along the second principal axis and the connection structure as well as the ID estimates indicate a two dimensional surface. For 40 nodes (D01 level) respectively 60 nodes and more (D10 level) the distribution of pointers begins to approximate the noise and hence the ID estimate drifting to three. This is also indicated in figure 6 showing a first phase of ID estimation one, a phase transition to ID estimation of two and a final transition to ID estimation of three. As expected, the ID-1 and ID-2 periods last longer on D10 level than on D01 level.

The data set unequivocally demonstrates that it does not make sense to speak of *the* intrinsic dimensionality of a noisy data set and to attempt to design a non-interactive algorithm just returning this number. Whether the data set has ID one, two or three cannot be decided on basis of the data alone. All three interpretations are perfectly correct. We need additional information, i.e. the scale or resolution to look at the data. Our ID estimation procedure starts on the coarsest resolution and constantly refines it. It is the users task to select the appropriate scale and the final ID estimate based on prior knowledge or his subjective bias.

Taking a closer look at the influence of noise, the $OTPM$ of 60 nodes and associated pointers has a mean squared quantization error of 0.23. With similar arguments as for the previous example we note that discrimination

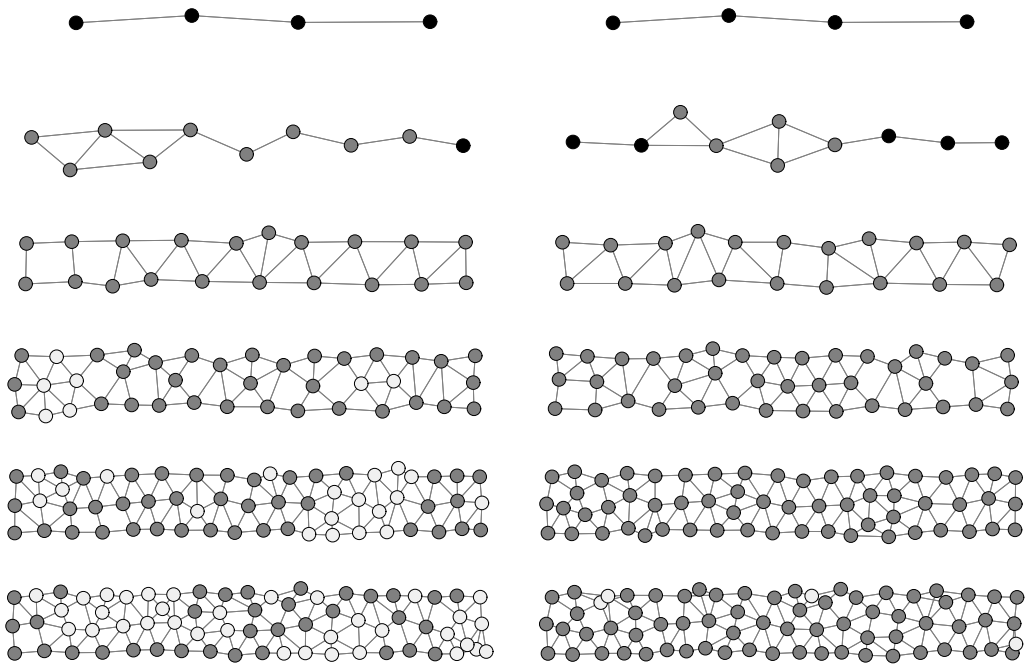

Figure 5: Local ID estimation for the rectangular data set for a growing number of pointers (nodes in the $OTPM$) on the D01 (left column) and D10 level (right column). From top to bottom: 4, 10, 20, 40, 60, 70 nodes. Dark circles indicate a local ID estimate of one, medium dark circles an estimate of two and light circles of three dimensions.

between the second intra-surface eigenvector and the noise component would be impossible for this local region size with the usual local PCA approach.

4.3 Further Demonstrations

In order to get a further impression of how the ID-estimation procedure copes with non-linearities we here give an example of ID-estimation for data stemming from surfaces of d-dimensional hyperspheres. Each data set consists of 5000 uniformly distributed samples on the surface of the d-dimensional spheres in the first "octant". In case of noisy data uniform noise with an amplitude of as much as half the radius was added perpendicular to the

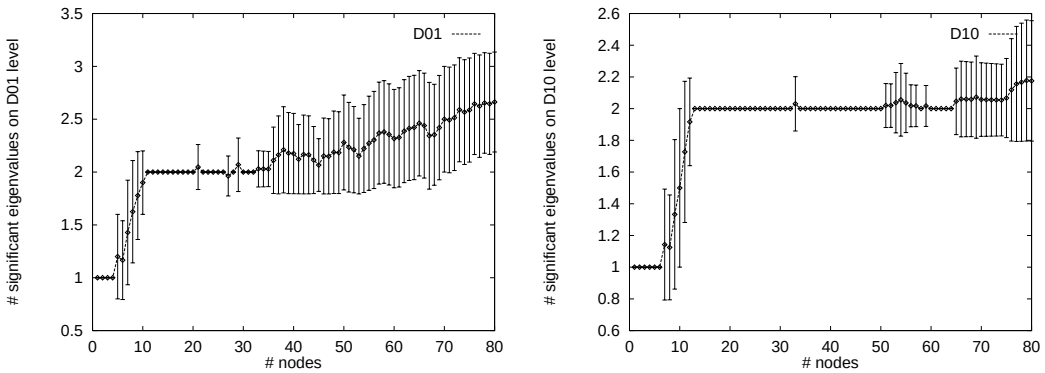

Figure 6: ID plot for the rectangular data set. Left: On D01 level, Right: On D10 level

surface.

Figure 7 shows the ID estimation procedure in progress for data from a 3d-hypersphere surface. For the noise-free case (left) estimation on the D05 and D10 level reveals the correct ID. The curvature is too high to give a clear hint on the D01 level. With noisy data (right) estimation on the D05 level gives no clear hint. On D10 level, however, an ID of two is correctly indicated from four to twenty nodes.

Figure 8 shows ID estimation for a 6d hypersphere surface. Correct estimates for the noise-free case (left) are indicated on the D05 and D10 level whereas estimation of the true ID on the D01 level would be difficult. For the noisy data set (right) ID estimation works perfectly well on both the D05 and D10 level. That actually we obtain better results in this case than for the noisy data of the 3d hypersphere surface is due to the increased surface area in 6 dimensions. Since the variance of the noise remains constant the ratio of intra-surface variance to noise variance increases and hence the incremental LBG stage placing more nodes in the surface.

As a final example involving artificial data let us regard some Lissajous figures in d dimensions generated by the vector valued function $f : R^1 \rightarrow R^n$ with

$$f_i(k) = 3 * \sin(2\pi k + i) \quad \text{with} \quad i = 1, \ldots, n-1 \tag{12}$$

and k randomly distributed in the interval $[0, 1]$. For noise-free data we had $f_n(k) = const$, else $f_n(k) = u$ with u denoting uniform noise with amplitude ± 0.5 (variance $1/12$). Hence data lie on a closed 1-d surface (ID = 1)

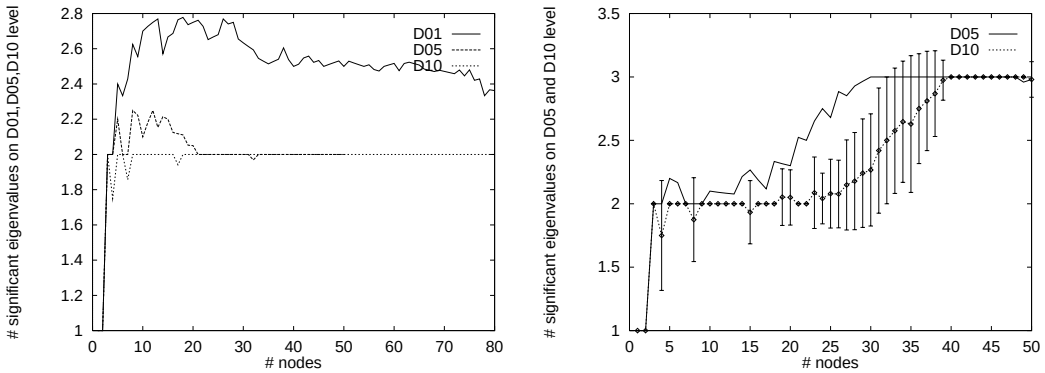

Figure 7: ID plots for the surface of a 3d hypersphere. Left: noise free data on D01, D05 and D10 level; Right: noisy data on D05 and D10 level including errorbars for the D10 level

embedded into an n-dimensional input space.

Figure 9 depicts the ID-estimation for noise-free 3-, 100- and 2500- dimensional data sets on the D10 level (left) and noisy 3- and 100-dimensional data sets on the D10 level (right). On this level the correct ID can be deferred for all noise-free data sets (left) but the plot is less conclusive for the noisy 3-d Lissajous figure (right). For both the noise-free and the noisy data sets, ID estimation appears to become easier with increasing dimension. This can again be explained with an increasing length of the line with growing dimensions. In the noise-free case it has the effect of decreasing the non-linearity and hence enabling ID-estimation with fewer pointers (larger local region sizes). In the noisy case the increased length of the line again causes a higher ratio of intra-surface variance and noise variance, hence diminishing the effect of noise.

4.4 ID estimation of image sequences

The experiment with the Lissajous figures paved the way for the application of our ID-estimator to image sequences. The sequences under investigation have been generated with one degree of freedom and hence they lie on a one dimensional trajectory in image space. The experiment with the Lissajous figures demonstrated that the task to estimate the ID from such a data set embedded in a very high dimensional input space does not pose a principal problem.

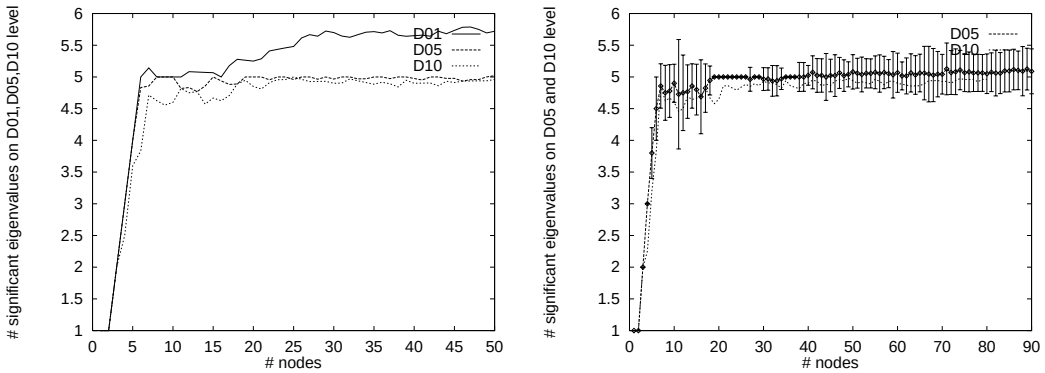

Figure 8: ID plots for the surface of a 6d hypersphere. Left: noise free data on D01, D05 and D10 level; Right: noisy data on D05 and D10 level including errorbars for the D10 level

4.4.1 Rotating grey ramp

The image sequence under investigation in this example has been generated by taking 180 snapshots (every 2°) with a resolution of 256×256 pixels of a robot rotating a cylinder around its z-axis (from 0° to 360°). Since the background remained constant, the images lie on a closed 1-dimensional trajectory in image space. In order to generate a smooth transition from image to image we wrapped a symmetrical grey ramp (256 grey values) around the cylinder. This grey ramp as well as three snapshots from the sequence are displayed in figure 10. The noise in the measurement process is approximately Gaussian with a standard deviation of 1.75 grey values per pixel.

ID-estimation on the D05 level (figure 11, left) indicates that the ID is at most 2 [7]. Estimation on the D10 level indicates an ID between 1 and 2 whereas estimation on the D20 level speaks for an intrinsic dimensionality of 1, the true ID. It is interesting to notice that in spite of the 65536-dimensional input space the ID-estimate never exceeds 2 on all three levels. The explanation, revealed by an analysis of the $OTPM$s for each number of nodes, is that the edges in the $OTPM$ actually form a (one dimensional) circle, i.e. the intrinsic structure (topology) is correctly represented by a one dimensional

[7] The reader should bear in mind that in this and the following experiment we do *not* try to estimate any properties of the objects in the scene, e.g. the shape of the cylinder, but the number of free parameters that generated the image sequence. Each image is just treated as one point in 65536d image space.

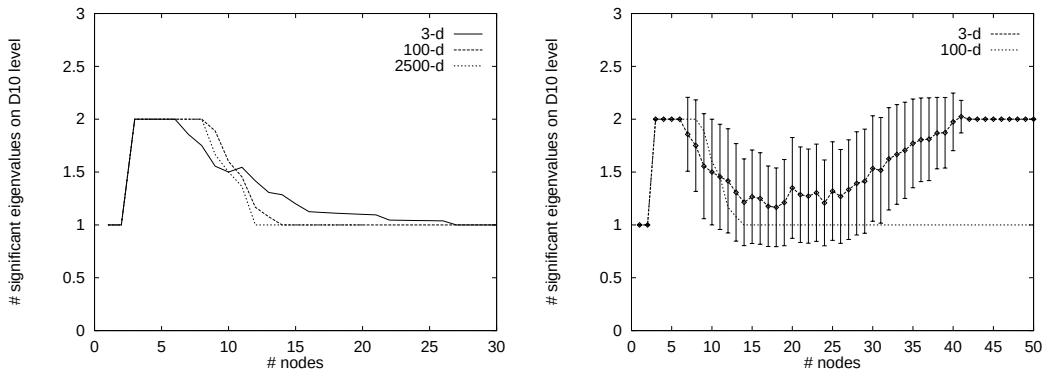

Figure 9: ID plots for Lissajous figures on D10 level. Left: noise free data of Lissajous figures in 3, 100 and 2500 dimensions; Right: noisy Lissajous figures in 3 and 100 dimensions

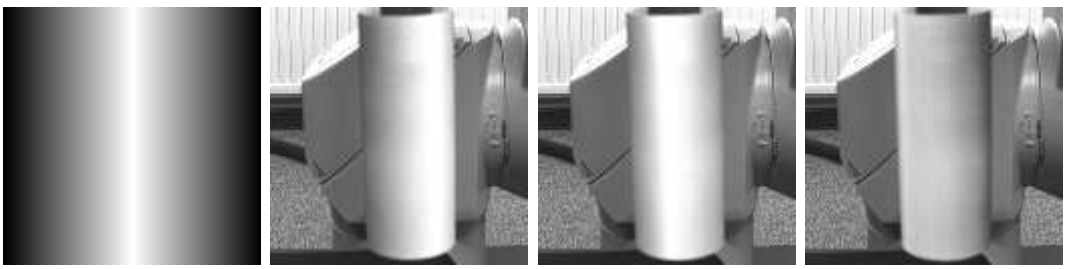

Figure 10: Grey ramp under different rotations. From left to right: Original (symmetric) grey ramp, grey ramp wrapped around a bottle with part of the robot arm in the background under 0°, 45°, 90° rotation

graph. Due to noise and the non-linearity of the trajectory, however, the local PCA taking the two difference vectors of a pointer and its two topological neighbors as input, does not indicate a one dimensional local structure on each level.

We have also performed ID estimation for the same sequence of images on a reduced image resolution of 64×64 pixels, obtained from the original one by averaging over a local neighborhood of 4×4 pixels. The result (figure 11, right) is similar to that of the full scale sequence except that we get slightly better estimates on D10 level. We attribute this to the noise reduction property of averaging over the 4×4 windows.

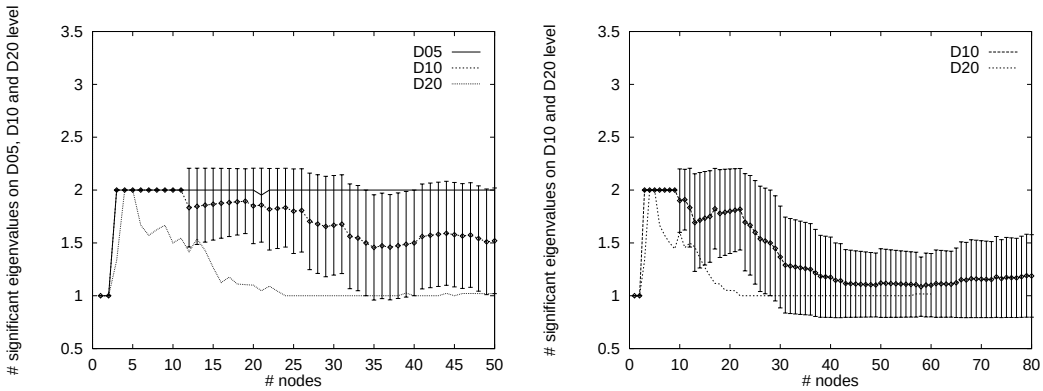

Figure 11: ID plots for rotating grey ramp. Left: For full scale 256×256 image sequence on D05, D10 and D20 level with errorbars for the D10 level; Right: For reduced scale 64×64 image sequence on D10 and D20 level with errorbars for the D10 level

4.4.2 Rotating bottle

As a last test we applied the ID estimation procedure to an image sequence of 180 images with a resolution of 64×64 pixels obtained by rotating a bottle of beer by 360° and taking a picture every 2° (see figure 12).

Figure 12: Beer bottle under different rotations with robot arm in the background. From left to right: Under 0°, 45°, 90° and 135° rotation

ID-estimation (figure 13) reveals similar results for all three levels (D05, D10 and D20) and indicates an ID of two. This again is quite impressive taking into account the 4096d input space. However, it is one more than the true ID. The answer is revealed by analyzing the $OTPM$s. Similar to the experiment with the rotating grey ramp, the edges in the $OTPM$ form a circle and hence have grasped the intrinsic one-dimensional structure of the

data [8]. However, in spite of successive pointers being neighbors with respect to the Euclidean metric, each two successive difference vectors which serve as input to the local PCA are highly uncorrelated. This can be understood by considering a rotating black/white bar. Three images (p_1, p_2, p_3) taken under successive rotations are neighbored w.r.t. the L_2 norm but the two difference images ($p_1 - p_2$ and $p_3 - p_2$) are completely uncorrelated ($(p_1 - p_2)^T (p_3 - p_2) = 0$) [9]. Since they have about the same amplitude, local PCA will always detect two significant eigenvalues.

Our method is not the only ID estimation technique suffering from difficulties with non-continuous jumps in the data (as revealed by sequences of b/w images). They actually represent an ill-posed problem for any ID-estimator. The problem can be coped with by smoothing. In the case of our rotating bottle image sequence low pass filtering of the images would increase the correlation between neighbored pixels and thus between successive images and lead to the correct ID estimate.

5 Related Work

In this section we want to relate our approach to previous work limiting our discussion to the most closely related local approaches. For an introduction to global ID estimation methods see e.g. [3], for a critical evaluation of different ID estimation algorithms see [24].

The algorithm most closely related to ours is that of Fukunaga and Olsen, [8]. It is based on local PCAs in local regions (overlapping hyperspheres) of varying size and uses the same significance criterion Dα as we do (eq. 9). Instead of plotting the ID-estimate over the local region size the results for different region sizes and values of α are summarized in histograms. The algorithm does not attempt to extract a representation capturing the intrinsic structure (topology) of the data set. By using straight forward PCA it has cubic complexity in the input dimensionality and application to ID estimation of e.g. full-scale images is clearly out of range. Also, by performing local PCA directly on the data, the influence of noise is much more severe than in our ID-estimation procedure which attempts to filter out the noise by a clustering

[8] With just a simpe test (check, if a node has only two neighbors and these neigbors are not connected) local ID estimation could stop here with output "1".

[9] This effect is known as the *whitening effect* of the difference operation.

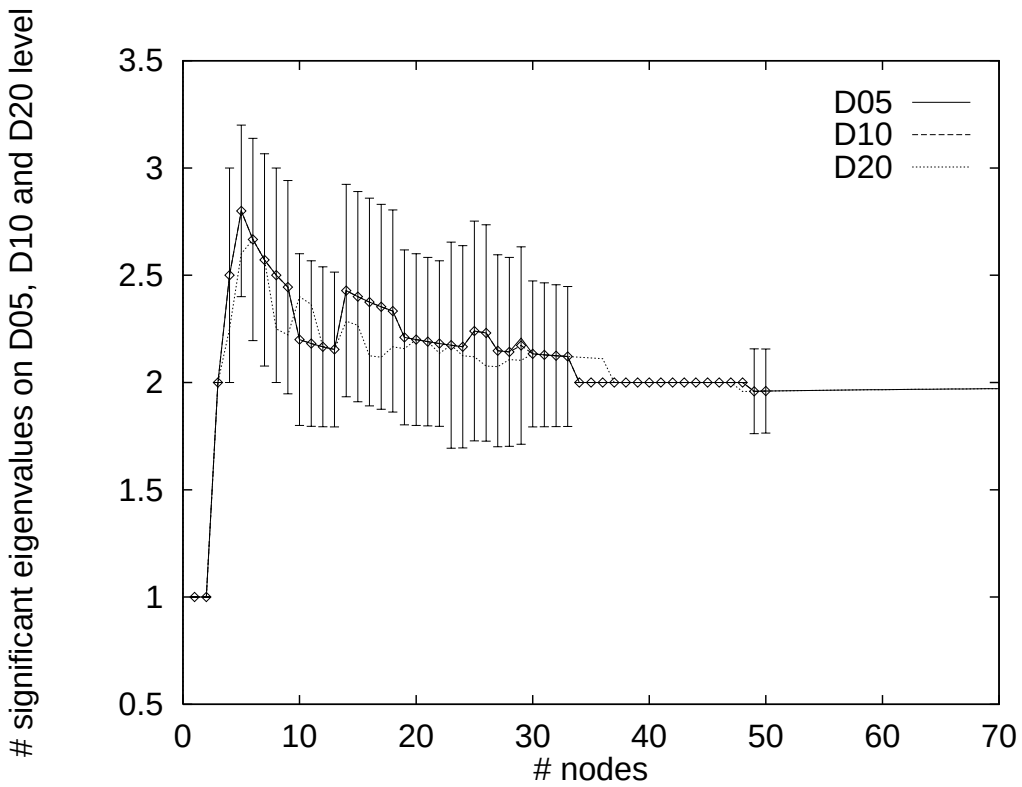

Figure 13: ID plots for rotating beer bottle on D05, D10 and D20 level with errorbars for the D10 level. The D05 and D10 plots coincide

stage and takes the cluster centers as input to the local PCA. We have demonstrated this increased discrimination ability of our method in sections 4.1 and 4.2. In their comparative study on ID estimation algorithms, [24], Wyse et al. found Fukunaga and Olsen's algorithm to perform "reasonably well" on a variety of data sets and to be one of the most reliable and easy to use methods. We conjecture that our ID-estimation procedure which from a users point of view can be regarded as an enhanced version of Fukunaga and Olsen's algorithm in terms of speed, accuracy and usability would at least have got the same predicate. In a more recent study,[10], Verveer and Duin compared slightly modified versions of Fukunaga and Olsen's algorithm and Pettis local ID estimator,[7]. Both algorithms performed well, the main drawback of Pettis algorithm being its trend to underestimate and suffering from "edge effects" and the main drawback Fukunaga and Olsen's algorithm

being an increased sensitivity to noise and quickly increasing computing time with dimensionality of input space - just the problems we have "fixed" in this paper.

The idea of using $OTPM$s for ID estimation is not entirely new but has been used before by Frisone et al., [4]. They tried to directly use the $OTPM$ for ID estimation by relating the number of edges emanating from the nodes in the $OTPM$ to the kissing number. As discussed in section 2.1 this approach suffers from both practical and theoretical pitfalls. On the practical side, the approach needs very large data sets and heavily relies on near optimal placement by a potentially slow clustering procedure. On the theoretical side, just too little is known about quantizers and corresponding kissing numbers. However, the idea of directly using the $OTPM$ for ID estimation (without additional PCA) is surely worth further pursuit. In case of the turning beer bottle image sequence, for instance, the $OTPM$ had grasped the correct topological structure but local PCA failed to reveal it.

Kambathatla and Leen, [25], have developed an algorithm for fast non-linear dimension reduction. Although not primarily intended for ID estimation it works similar to our procedure in that it builds a local linear model of the data by merging local PCA with clustering. Data is first clustered into N clusters, then local PCA is used in each Voronoi cell to obtain m_i eigenvectors. Together with the centroid of the corresponding cells the eigenvectors are then used for linear approximation of the data set. Obviously, their procedure could benefit from ideas presented in this paper, i.e. the additional construction of an $OTPM$ and using it for efficient local PCA in the same way we do. As in our case, this would lead to only a linear complexity of the local PCAs. Vice versa, the work of Kambathatla and Leen shows the straight forward way to use the representation we extract in course of ID estimation (cluster centers and eigenvectors) for auto association and vector quantization (by means of a linear approximation of the data set). Since Kambathatla and Leen's results are quite encouraging we conjecture that the results obtainable by the extended ID-estimation procedure will be as well.

6 Conclusion

We have presented an algorithm for estimating the intrinsic dimensionality of low dimensional submanifolds embedded in high dimensional feature spaces. The algorithm belongs to the category of local ID-estimation procedures, is based on local PCA and directly extends and improves its predecessor, the algorithm of Fukunaga and Olsen, [8], in terms of computational complexity and noise sensitivity. The main ideas are first to cluster the data, second to construct an $OTPM$ and third to use the $OTPM$ and not the data itself for local PCA.

Clustering is responsible for an even distribution of the cluster centers and for noise reduction, i.e. placing the centers in the manifold. The local PCA taking difference vectors of centers as an input benefits from the noise reduction property of the clustering stage. Its output, the eigenvalues, give a better hint at the local ID than those of straight forward local PCA on the data itself always including the full variance of the noise.

Constructing the $OTPM$ for the cluster centers provides a low dimensional representation of the data which optimally reflects the intrinsic (topological) structure of the data. Independent of the dimension of the input space and invariant w.r.t. scaling and rigid transformations it provides an ideal basis for ID estimation. Exploiting the $OTPM$ for local PCA, our ID estimation procedure has only linear time complexity in the dimension of the input space and the invariance properties directly transfer to the estimate. We conjecture that more direct use of the $OTPM$ offers a possibility to improve the ID-estimates. For instance it is trivial to decide whether an $OTPM$ (a graph) has one dimensional structure or not.

Besides tests on a variety of illustrative artificial data sets the procedure was applied to ID-estimation of image sequences with image resolutions of up to 256×256 pixels. Such application is out of reach for conventional ID-estimation procedures based on local PCA and to the best of our knowledge has not been tackled before.

$OTPM$s together with eigenvectors and eigenvalues returned by local PCA are not only useful for ID estimation but can be used for linear approximation of the data and construction of auto-associators in quite an obvious way. Such associators will work by projecting new data to the local subspaces spanned by the eigenvectors, i.e. by projecting to the linear approximation of the manifold. Extension to the construction of hetero-associators working

on basis of the same principal needs only one more little step (using e.g. generalized radial basis functions). Application to visual learning and recognition of objects from appearance as pioneered by Murase and Nayar, [16], appears to be straight forward as well and closes this brief summary of our near-future work.

References

[1] B. Mandelbrot, *Die fraktale Geometrie der Natur.* Birkhaeuser Verlag, Basel, 1991.

[2] A. Heyting and H. Freudenthal, *Collected works of L.E.J. Brouwer.* North Holland Elsevier, 1975.

[3] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Prentince Hall, 1988.

[4] F.Frisone, P.Morasso, F.Firenze, and L.Ricciardiello, "Application of topology-representing networks to the estimation of the intrinsic dimensionality of data," in *Proc. of the International Conference on Artificial Neural Networks*, vol. 1, pp. 323–327, 1995.

[5] R. S. Bennett, "The intrinsic dimensionality of signal collections," *IEEE Transactions on Information Theory*, vol. 15, pp. 517–525, 1969.

[6] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1–27, 1964.

[7] K. Pettis, T. Bailey, T. Jain, and R. Dubes, "An intrinsic dimensionality estimator from near-neighbor information," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, vol. 1, pp. 25–37, 1979.

[8] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Transactions on Computers*, vol. 20, no. 2, pp. 176–183, 1971.

[9] G. V. Trunk, "Statistical estimation of the intrinsic dimensionality of a noisy signal collection," *IEEE Transactions on Computers*, vol. 25, pp. 165–171, 1976.

[10] P. J. Verveer and R. P. Duin, "An evaluation of intrinsic dimensionality estimators," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, vol. 17, no. 1, pp. 81–86, 1995.

[11] T. Martinetz and K. Schulten, "Topology representing networks," in *Neural Networks*, vol. 7, pp. 505–522, 1994.

[12] T. Villmann, R. Der, and T. Martinetz, "A novel aproach to measure the topology preservation of feature maps," *ICANN*, pp. 289–301, 1994.

[13] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups.* Grundlehren der mathematischen Wissenschaften 290, Springer Verlag NY, 1988.

[14] T. Martinetz, *Selbstorganisierende neuronale Netzwerkmodelle zur Bewegungssteuerung.* PhD thesis, Physik-Department TU Muenchen, 1992.

[15] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C - The Art of Scientific Computing.* Cambridge University Press, 1988.

[16] H. Murase and S. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[17] P. L. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 139–149, 1982.

[18] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transaction on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[19] A. Gersho and M. Gray, *Vector Quantizers and Signal Compression.* Kluwer Academic Publishers, 1992.

[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the Fith Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, Berkeley: University of California Press, 1967.

[21] T. Kohonen, "The neural phonetic typewriter," *Computer*, vol. 21, pp. 11–24, 1988.

[22] H. Ritter and K. Schulten, "On the stationary state of kohonen's selforganizing sensory mapping," *Biological Cybernetics*, vol. 54, pp. 99–106, 1986.

[23] H. U. Bauer, "Controlling the magnification factor of self-organizing feature maps," *Neural Computation*, vol. 8, pp. 757–771, 1996.

[24] N. Wyse, R. Dubes, and A. Jain, "A critical evaluation of intrinsic dimensionality algorithms," in *Pattern Recognition in Practice* (E. Gelsema and L. Kanal, eds.), pp. 415–425, North-Holland Publishing, 1980.

[25] N. Kambhatla and T. Leen, "Fast non-linear dimension reduction," in *Advances in Neural Information Processing Systems, NIPS 6*, pp. 152–159, 1994.