Local PSOMs and Chebyshev PSOMs Improving the Parametrised Self-Organizing Maps[1]

Jörg Walter and **Helge Ritter**

Department of Information Science
University of Bielefeld, D-33615 Bielefeld, FRG
Email: {walter,helge}@techfak.uni-bielefeld.de

Abstract

We report on two new improvements for the "Parameterised Self-Organizing Map" (PSOM). Both achieve a significant increase in mapping accuracy and computational efficiency.

For a growing number of training points the use of higher order polynomials to construct the PSOM "mapping manifold" in [7] can suffer from the increasing tendency to oscillate between the support points. We propose here to confine the algorithm to a subset of the training knots, resulting in what we call the "local-PSOM" algorithm. This allows to avoid the use of high-degree polynomials without sacrificing accuracy. At the same time, the new approach offers a significant saving in required computations.

A second way to improve the mapping preciseness makes use of the superior approximation properties of Chebyshev polynomials for the PSOM mapping manifold.

The benefits of the two new approaches are demonstrated with two benchmark problems: (i) approximating a Gaussian bell function and (ii) learning of the (forward and inverse) kinematics of a 3 DOF robot finger. In both cases the PSOM algorithm exhibit an excellent generalisation ability and that already for a very small training set size of $3\times3\times3$ points.

1 Introduction

In the past, the Self-Organizing Map (SOM) has proven to be a very versatile approach for constructing data mappings in many different domains [3, 9]. While the original SOM is of a discrete nature, continuous version – the "Parametrised Self-Organizing Map" (PSOM) has been proposed recently [7] and studied in the context of learning tasks for robotics and vision [8, 12, 10]. It was shown that PSOMs exhibit a number of attractive features, among them *rapid learning* (as a result of a reduced number of adaptable parameters), an *associative completion* capability for continuous-valued partial input vectors, and the possibility of *"factoring"* higher-dimensional mappings into a hierarchy of lower-dimensional PSOMs ("Meta-PSOMs").

In the present paper, we present two significant improvements to the PSOM-algorithm: a *"Chebyshev PSOM"* – an improved knot-spacing scheme leading to higher approximation accuracy, and a *"local-PSOM"* that is computationally more efficient than the

[1]This work was supported by the German Ministry of Research and Technology (BMFT), Grant No. ITN9104AO. Any responsibility for the contents of this publication is with the authors.
In *Proc. ICANN'95*, pages 95–102, Paris October 1995

standard PSOM. These improvements are studied for two different tasks: learning to approximate a 2D-Gaussian bell function, and learning the complete kinematics of a 3-jointed robot finger. The second mapping has a region where it becomes almost singular and so provides a very challenging benchmarking task for the algorithm.

2 How the Standard PSOM works

This section gives a very brief summary of the standard PSOM algorithm, derived from the "Self-Organizing Map" (SOM) algorithm (for more details, see [7, 11]).

In contrast to a standard SOM, which consists of a discrete grid $\mathbf{A}$ of formal neurons each labeled by an index $\mathbf{a} \in \mathbf{A}$ and described by a "reference vector" $\mathbf{w_a}$, a PSOM consists of a continuous manifold M, parameterised by a continuous variable $\mathbf{s} \in S$ and described by a smooth function $\mathbf{w}(\mathbf{s})$. In both cases, $\mathbf{w_a}$ and $\mathbf{w}(\mathbf{s})$ take their values in an embedding space X, from which the input vectors $\mathbf{x}$ are also drawn. The response of a SOM to an input $\mathbf{x} \in X$ is determined by the reference vector $\mathbf{w_{a^*}}$ of the discrete "best-match" node $\mathbf{a}^* = \text{argmin}\, \|\mathbf{w_{a'}} - \mathbf{x}\|$, where $\mathbf{a}' \in \mathbf{A}$. Similarly, the response of a PSOM is determined by the value $\mathbf{w}(\mathbf{s})$ at the continuous best-match location $\mathbf{s}^* \in S$, defined by the analogous Eq. 1 below.

Frequently, it is useful to view the embedding space X as the product of some "input" subspace X^{in} and some "output" subspace X^{out}. The distance $dist(\cdot)$ is used for finding the best-match location $\mathbf{s}^* \in S$

$$\mathbf{s}^* = \text{argmin}\; dist\,(\mathbf{w}(\mathbf{s}), \mathbf{x}) \tag{1}$$

and is usually the Euclidean norm, applied to the input components of $\mathbf{x}$. Then $\mathbf{w}(\mathbf{s}^*)$ can be viewed as an associative completion of the input space component of $\mathbf{x}$. In other words, the function $dist(\cdot)$ actually selects the input subspace X^{in} since for the determination of $\mathbf{s}^*$ (and as a consequence, of $\mathbf{w}(\mathbf{s}^*)$) only those components of $\mathbf{x}$ matter, that are declared in the distance metric $dist(\cdot)$. As an important feature, $dist(\cdot)$ can be changed on demand, allowing e.g. to reverse the mapping direction using the same PSOM.

How can the required smooth manifold $\mathbf{w}(\mathbf{s})$ be constructed? In principle, there are several ways, but a particular close analogy to the standard SOM results when we take

$$\mathbf{w}(\cdot):\; S \to M \subset X, \qquad \mathbf{w}(\mathbf{s}) = \sum_{\mathbf{a}\in\mathbf{A}} H(\mathbf{a}, \mathbf{s})\, \mathbf{w_a} \tag{2}$$

This means that, we need a *"basis function"* $H(\mathbf{a}, \mathbf{s})$ for each formal neuron (in the following also called "knot"), weighting the contribution of its reference vector ("training point") $\mathbf{w_a}$ depending on the location $\mathbf{s}$ relative to the knot position $\mathbf{a}$, and possibly, also *all* other knots (however we drop in our notation the dependency $H(\mathbf{a}, \mathbf{s}) = H(\mathbf{a}, \mathbf{s}; \mathbf{A})$ on the latter.)

Specifying $\mathbf{a}$ introduces a topological order between the training points $\mathbf{w_a}$: training vectors assigned to neighboring knots $\mathbf{a}, \mathbf{a}' \in \mathbf{A}$ are recognized to be neighbors. This allows the PSOM to draw extra curvature information from the training set, information which is not available within other techniques, such as the Radial Basis approach (e.g. [5, 2]).

The topological organisation of the given data points is crucial for a good generalisation behavior. For a general data set the topological ordering of its points may be quite irregular and a set of suitable basis functions $H(\mathbf{a}, \mathbf{s})$ difficult to construct. A suitable set of basis functions can be constructed in many ways but must meet two conditions: (i) $H(\mathbf{a}, \mathbf{s})$ must be *ortho-normal* $H(\mathbf{a}_i, \mathbf{a}_j) = \delta_{ij}$ $(\forall\ \mathbf{a}_i, \mathbf{a}_j \in \mathbf{A})$ to make the manifold M passing through all supporting knots; (ii) Division of unity: $\sum_{\mathbf{a}} H(\mathbf{a}, \mathbf{s}) = 1,\ \forall \mathbf{s}$ (consider the task of mapping a constant function $\mathbf{x} = \mathbf{w}_{\mathbf{a}}$. Obviously the sum of basis functions should be flat as well, which means, the sum of all contribution weights should be one.)

A simple construction of basis functions $H(\mathbf{a}, \mathbf{s})$ becomes possible when the topology of the given points is sufficiently regular. A particularly convenient situation arises for the case of a multidimensional rectangular grid. In this case, the set of functions $H(\mathbf{a}, \mathbf{s})$ can be constructed from products of one-dimensional Lagrange interpolation polynomials, as illustrated in the following and also described in the Appendix.

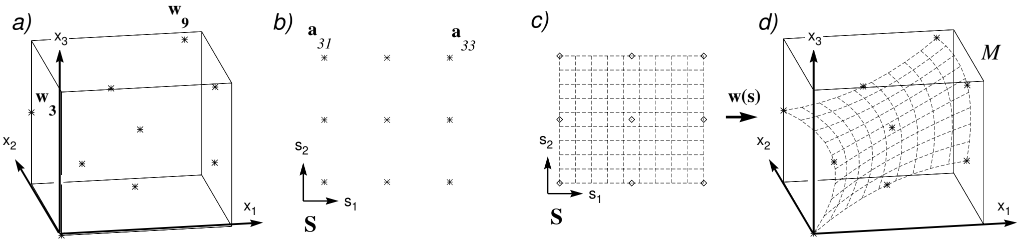

Figure 1: a–d: Illustration of the training setup for a 3×3 PSOM for the simple task of a linear mapping defined by a tilted plane in 3D (note that a 2×2 PSOM would be sufficient). *(a)* The nine training points are roughly topologically ordered, as shown by the tick marks in a orthonormal projection of the 3D embedding space X and *(b)* their corresponding auxiliary mapping coordinates $\mathbf{a}$ in the mapping manifold S, lying on a rectangular grid. By the means of Eq. 2, S spans an $m = 2$ dimensional manifold M in the embedding space X, visualised in *(c–d)*. Note, the training set does not necessarily lie on any exact grid in X. The cube is drawn for visual guidance only.

Figures 1 illustrates the training set up of a PSOM that represents a $m = 2$ dimensional "mapping manifold" S spanning an "embedded manifold" M in $X \subset \mathbb{R}^3$. For this pedagogic example, we choose M as a tilted plane in which 9 sample points $\mathbf{w}_1, \ldots, \mathbf{w}_9$ are assumed to be given. To construct the PSOM requires assignment of the location $\mathbf{a}$ in the mapping manifold S to each given training point $(\mathbf{w}_1 = \mathbf{w}_{\mathbf{a}_{11}}; \ldots; \mathbf{w}_9 = \mathbf{w}_{\mathbf{a}_{33}})$, thus specifying the topological organization, here as the shown rectangular 3×3 grid $\mathbf{A}$.

The discrete best-match search in the standard SOM is now replaced by solving the continuous minimization problem for the determination of $\mathbf{s}^*$. A simple approach to this is to first do the (SOM-like) discrete best-match search to find $\mathbf{s}_{start} = \mathbf{a}^*$ in the knot set $\mathbf{A}$, followed by an iterative gradient descent for Eq. 1. However, for good results a efficient implementation of the gradient descent is important. We found the Levenberg-Marquardt scheme best suited, details must be reported elsewhere [11]. Likewise, the possibility of adapting a PSOM according to a Kohonen-type learning rule will not be considered here.

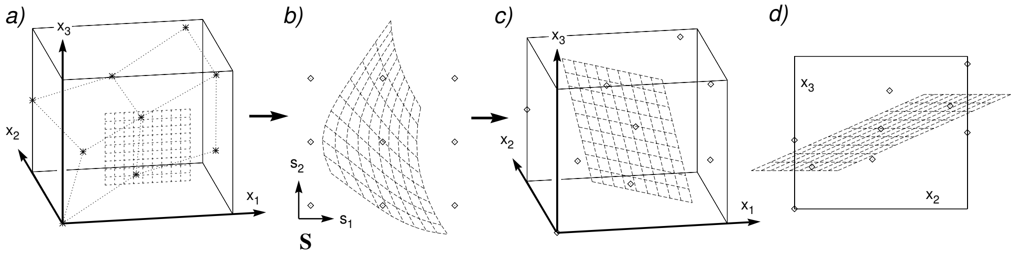

Figure 2: a–d: PSOM recall procedure for a rectangular spaced set of 10×10 (x_1, x_3) tuples to (x_2, x_3), together with the original training set: *(a)* the input space in the $x_2 = 0$ plane, *(b)* the resulting (by Eq. 1) mapping coordinates $\mathbf{s} \in S$, *(c)* the completed data set in X , *(d)* the desired output space projection (looking down x_1).

Fig. 2b shows the result of the "best-match projection" $\mathbf{x} \mapsto \mathbf{s}^*(\mathbf{x})$ into the manifold M, when $\mathbf{x}$ varies over a regular 10×10 grid in the plane $x_2 = 0$ and x_1, x_3 is selected as the input space (by defining $dist(\cdot)$ appropriately). Fig. 2c–d shows a rendering of the associated "completions" $\mathbf{w}(\mathbf{s}^*(\mathbf{x}))$ which form a grid in X.

3 Extensions to the PSOM algorithm

When an increase in mapping accuracy is desired, one normally increases the number of training points per parameter axis. Here we encounter two shortcomings with the original approach: (i) The basis polynomials exhibit unsatisfactory convergence properties with increasing order. Mappings of sharply peaked functions can force a high degree interpolation polynomial to increasingly strong oscillations, spreading in between the support knot points of the entire manifold. (ii) The computational effort per mapping manifold dimension grows by $O(n^2)$ for the number of training points n in each axis. Even with a moderate number of sampling points along each parameter axis, the inclusion of all nodes in Eq. 2 may still require too much computational effort if the dimensionality of the mapping manifold m is high (say, $m > 4$).

3.1 The "Local" PSOMs

Both aspects motivate an important extension to the standard PSOM approach. The basic idea is to dynamically construct the PSOM only on a sub-grid of the full training data set. This sub-grid is (in the simplest case) always centered at the reference vector $\mathbf{w}_{\mathbf{a}^*}$ that is closest to the current input $\mathbf{x}$. The use of the sub-grid leads to lower-degree polynomials for the basis functions and involves a considerably smaller number of points in the sum in Eq. 2. Thus, the resulting *"local" PSOMs* ("l-PSOMs") provide an attractive scheme that overcomes both of the shortcomings pointed out in the previous paragraph. Fig. 3a–d explain the procedure.

3.2 Chebyshev spaced PSOMs

An alternative way to deal with the shortcomings addressed before is the use of an improved scheme for selecting the grid point spacing in the mapping manifold.

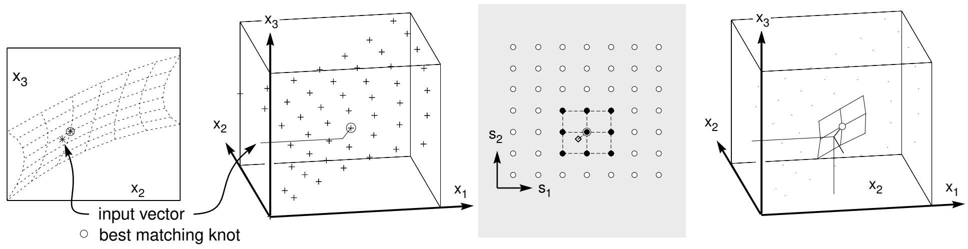

Figure 3: a–d: The example task of Fig. 1, but this time using a 3×3 local PSOM of a 7×7 training set. *(a–b)* The input vector (x_2, x_3) selects the closest knot $\mathbf{a}^*$ (in the now specified input space). The associated 3×3 knot sub-grid is indicated in *(c)*. The minimization procedure starts at its center $\mathbf{s} = \mathbf{a}^*$ and uses only the PSOM constructed from the 3×3 sub-grid. *(d)* displays the mapping result $\mathbf{w}(\mathbf{s}^*)$ in X, together with the selected sub-grid of knots in orthonormal projection. The light dots indicate the full set of training knots. (For the displayed mapping task, a 2×2 PSOM would be appropriate; the 7×7 grid is for illustrative purpose only.)

As is well-known from approximation theory, a equidistant grid point spacing is usually not the optimal choice for minimizing the approximation error (see [1, 6]). For polynomial approximation on an interval [-1,1], one should choose the support points a_j in each axis of $\mathbf{A}$ at locations given by the zeroes a_j of the Chebyshev polynomial $T_n(a)$

$$T_n(a) = \cos(n \text{ arccos } a)$$

$$\text{the zeros given by} \quad a_j = \cos\left(\pi(j - \tfrac{1}{2})/n\right). \tag{3}$$

It turns out that this choice of support points can be adopted for the PSOM approach without any increase in computational costs. As we will demonstrate shortly, the resulting *Chebyshev spaced* PSOM ("C-PSOM") tends to achieve considerably higher approximation accuracy as compared to *equidistant spaced* PSOMs for the same number of knot points. As a result, the use of Chebyshev PSOMs allows a desired accuracy to be achieved with a smaller number of nodes, leading to the use of lower-degree polynomials together with a reduced computational effort.

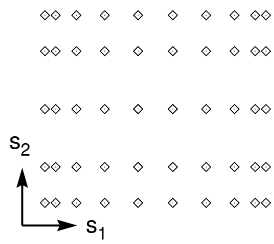

Figure 4: Placement of 5×10 knots $\mathbf{A} \in S$, placed according to the zeros of the Chebyshev polynomial in Eq. (3).

4 Results

4.1 Example: The Gaussian Bell

As a first example to compare the local and Chebyshev spaced PSOMs we consider the Gaussian bell function $x_3 = \exp\left(-\lambda^{-2}(x_1^2 + x_2^2)\right)$, with $\lambda = 0.5$ chosen to obtain a "medium sharp" curved function in the square region $[-1, 1]^2$. Using $n \times n$ (in x_1, x_2) equidistantly sampled training points we compute the root mean square deviation (RMS) between the goal mapping and the mapping of *(i)* a PSOM with equidistantly spaced knots, *(ii)* local PSOMs with sub-grid sizes $n' = 2$, 3, 4 (sub-grids use here also equidistant knot spacing), and *(iii)* PSOMs with Chebyshev spaced knots.

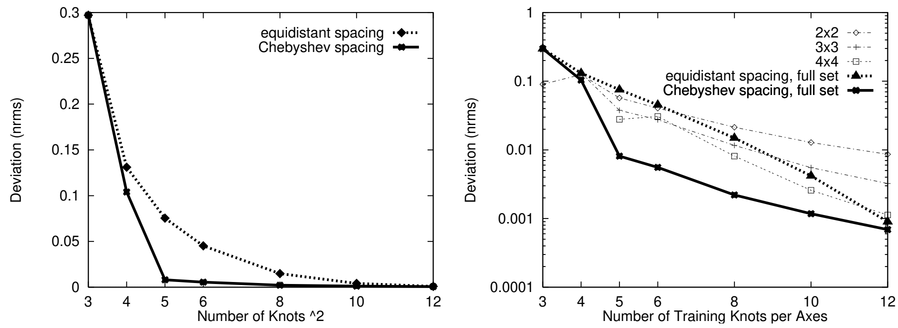

Figure 5: a–b; Mapping Accuracy of the Gaussian bell function for the presented PSOM variants – in a linear *(a)* and a logarithmic plot *(b)* – versus n the number of training points per axes.

Fig. 5 compares the numerical results (obtained with a randomly chosen test set) versus n. All curves show an increasing mapping accuracy with increasing number of training points, however, for $n > 3$ the Chebyshev spaced PSOM *(iii)* shows a significant improvement over the equidistant PSOM (for $n = 3$, the PSOM and the C-PSOM coincide, because the Chebyshev polynomials are always symmetric to zero and here equidistant as well).

For $n = 5$ the graphs show the largest differences. Fig. 6 zooms in, to detail the mapping performance as grid surface plots.

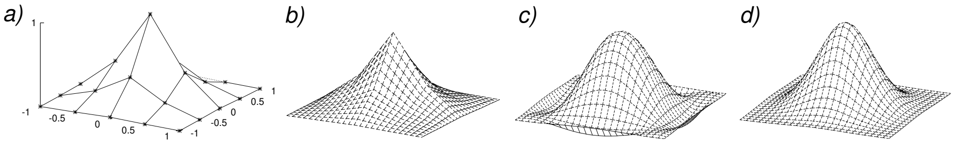

Figure 6: a–d, $n = 5$; *(a)* 5×5 training points are equidistantly sampled for all PSOMs; *(b)* shows the resulting mapping of the local PSOM with sub-grid size 2×2. *(c)* There are little overshoots in the marginal mapping areas of the equidistant spaced PSOM *(i)* compared to *(d)* the mapping of the Chebyshev-spaced PSOM *(ii)* which is for $n = 5$ already visually identical to the goal map.

4.2 Application: Robot Finger Kinematics

This section presents the results of applying the PSOM algorithm to the task of learning the kinematics of a 3 degree-of-freedom (DOF) robot finger, taken from a three-fingered modular hydraulic robot hand, developed by the Technical University of Munich [4]. The finger is actuated by spring-loaded oil cylinders driven by a remote "base station" that provides the hydraulic pressure. Its mechanical design allows roughly the mobility of the human index finger, scaled up by 10%. A cardanic base joint (2 DOF) offers sidewards gyring of $\pm 15°$ and full adduction with two additional coupled joints (1 DOF). See Fig. 7.

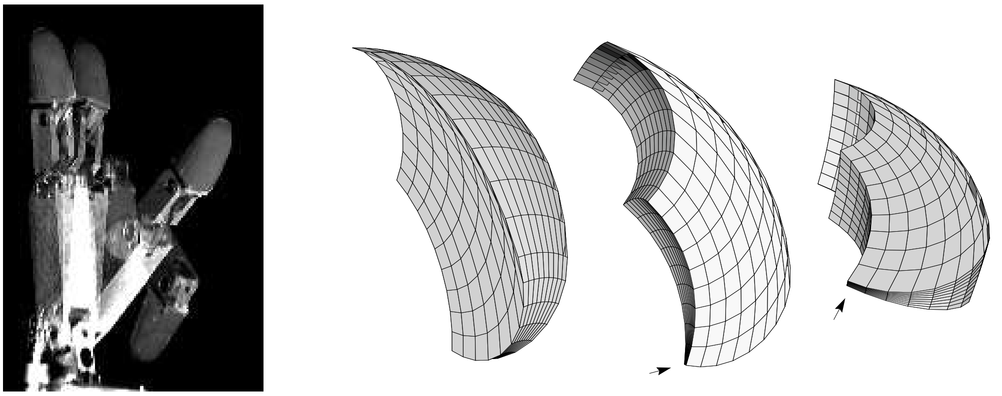

Figure 7: a–d: *(a)* stroboscopic image of one finger in a sequence of extreme joint positions. *(b–d)* Several perspectives of the workspace envelope $\vec{r}$, tracing out a cubical $10\times 10\times 10$ grid in the joint space $\vec{\theta}$. The arrow marks the fully adducted position, where one edge contracts to a tiny line.

In robotics, the kinematics denotes the relationship between joint or actuator coordinates and the Cartesian position of a particular end effector location. In the case of our finger, there are several coordinate systems of interest, e.g. the joint angles, the cylinder piston positions, one or more finger tip coordinates, as well as further configuration dependent quantities, such as the Jacobian matrices for force/moment transformations. All of these quantities can be simultaneously treated in one single common PSOM; here we demonstrate only the hardest part, the classical inverse kinematics. When moving the three joints on a cubical $10\times 10\times 10$ grid within their maximal configuration space, the fingertip (or more precise the mount point) will trace out the "banana" grid displayed in Fig. 7 (confirm this workspace with your finger).

We exercised several PSOMs with $n\times n\times n$ nine dimensional data tuples $(\vec{\theta}, \vec{c}, \vec{r})$, where $\vec{\theta}$ denotes the joint angles, $\vec{c}$ the piston displacement and $\vec{r}$ the finger point position, all equidistantly sampled in $\vec{\theta}$. Fig. 8a–b depicts a $\vec{\theta}$ and an $\vec{r}$ projection of the smallest training set, $n = 3$.

To visualise the inverse kinematics ability, we ask the PSOM to back-transform a set of workspace points of known arrangement. In particular, the workspace filling "banana" set of Fig. 7 should yield a rectangular grid of $\vec{\theta}$. Fig. 8c–e displays the actual result. The distortions look much more significant in the joint angle space *(a)*, and the piston stoke space *(b)*, than in the corresponding world coordinate result $\vec{r}'$ *(b)* after back-transforming the PSOM angle output. The reason is the peculiar structure; e.g. in areas close to the tip a certain angle error corresponds to a smaller Cartesian deviation than in other areas.

When measuring the mean Cartesian deviation we get an already satisfying result of 1.6 mm or **1.0 %** of the maximum workspace length of 160 mm. In view of the extremely minimal training set displayed in Fig. 8a–b this appears to be a quite remarkable result.

Nevertheless the result can be further improved by supplying more training points

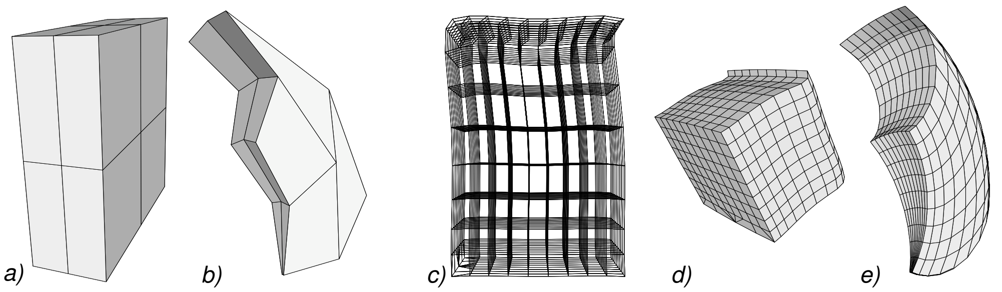

Figure 8: a–b and c–e; Training data set of 27 nine-dimensional points in X for the $3{\times}3{\times}3$ PSOM, shown as a perspective surface projections of the *(a)* joint angle $\vec{\theta}$ and *(b)* the corresponding Cartesian sub space. Following the lines connecting the training samples allows one to verify that the "banana" really possesses a cubical topology. *(c–e)* Inverse kinematic result using the grid test set displayed in Fig. 7. *(c)* projection of the joint angle space $\vec{\theta}$ (transparent); *(d)* the stroke position space $\vec{c}$; *(e)* the Cartesian space $\vec{r}'$, after back-transformation.

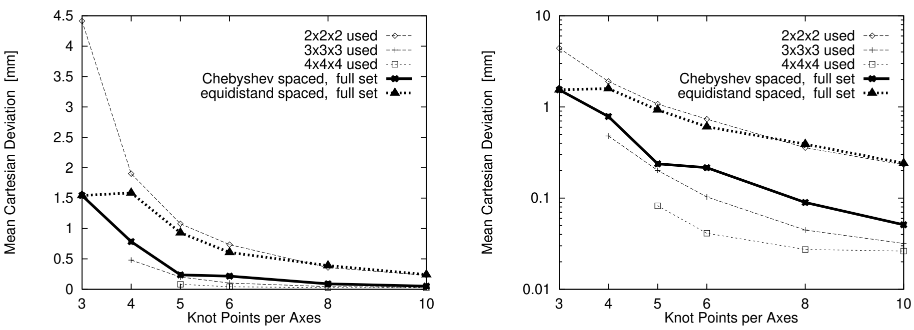

Figure 9: a–b: Mean Cartesian inverse kinematics error (in mm) of the presented PSOM types versus number of training knots per axes (using a test set of 500 randomly chosen positions; *(a)* linear and *(b)* log plot). Note, the result of Fig. 8c–e corresponds to the smallest training set $n = 3$. The maximum workspace length is 160 mm.

as shown in the asterisk marked curve in Fig. 9. The effective inverse kinematic accuracy is plotted versus the number of training knots per axes, using a set of 500 randomly (in $\vec{\theta}$ uniformly) sampled positions.

5 Discussion

How much do we gain with the Chebyshev and the local PSOM? Both Figs. 5 and 9 show that alone the use of the Chebyshev spacing can reduce the approximation error by a factor of roughly 4-10 for the present tasks, without incurring any additional computational costs. The explanation is the more graceful approximation behavior due to smaller extrema of the polynomial approximation in marginal areas of the mapping interval (see also [6]). The effect can be visually recognised in the little overshoots in Fig. 6c versus 6d.

However, in many cases the use of local PSOMs can be even more attractive, since they offer a significant reduction of the computational effort that still can even be accompanied with dramatic gains in accuracy. This is apparent in Fig. 9, where a 3×3×3 local PSOM outperforms a full set Chebyshev-PSOM in accuracy while at the same time requiring much less computation. If a $4{\times}4{\times}4$ local PSOM is used with a $8{\times}8{\times}8$ training set, the computational effort is comparable to that of a $4{\times}4{\times}4$ Chebyshev PSOM, but the average Cartesian error of 0.03 mm is much lower than the 0.8 mm of the latter case (or the 2 mm of a standard $4{\times}4{\times}4$ PSOM). Note also that even the 2×2×2 local PSOM almost reaches the same accuracy as the computationally much more expensive full set standard PSOM.

Of course, there may be functions that are locally less well described by a cubic polynomial, and the Gaussian bell function considered in Sec. 4.1. provides such an example. Here, the full-set Chebyshev PSOM remains superior, but a 4×4 PSOM is already sufficient to reach or outperform the accuracy of a full set equidistant PSOM while reducing the computational cost for $n = 12$ by a factor of almost one order of magnitude (Fig. 5). For $n = 12$, the accuracy of the 4×4 PSOM comes close to the Chebyshev performance.

Thus, rather low order local PSOMs combine good or even excellent accuracy with rather modest computational requirements. Their use preserves all other benefits of the standard PSOMs, such as associative completion and a high flexibility in choosing even very high-dimensional embedding spaces (not that the convergence behavior of the PSOM and the required number of nodes are not dominated by the embedding space dimensionality $X \subset \mathbb{R}^d$, but by the usually much lower intrinsic dimensionality of the manifold $S \subset \mathbb{R}^m$).

The placement of the sub-grid needs some attention concerning the continuity of the resulting map. Having an even number of sub-grid interval leave the ambiguity of placing the upper or lower of the two central intervals next to $\mathbf{x}$. For a detailed discussion see [11].

The concept of local PSOM bears some relations to the spline technique. Besides the fact that the PSOM algorithm can be also engaged as a general interpolation technique, share both the internal piecewise construction of a polynomial on a set of neighboring knots.

For the sub-grid size $n' = 2$ the local PSOM becomes actually identical to the multidimensional extension of the linear splines. The resulting mapping belongs to the “harmonic functions” and has interesting mathematical properties (e.g. “soap film membrane”, see [11]).

For $n' = 3$ and 4 its concept resembles the quadratic and the cubical spline, respectively. In contrast to the spline concept using piecewise assembled polynomials, we employ one single, dynamically constructed interpolation polynomials with the benefit of usability for multiple dimensions m.

We finally note an important point that might be easily overlooked. Unlike most interpolation schemes, the interpolation by a PSOM is not exclusively based on the metric distances between the data points in their embedding space X. Instead, the chosen grid $\mathbf{A}$ and labeling $\mathbf{a} \mapsto \mathbf{w_a}$ provide additional “topological” information to the PSOM that may be used to shape its generalization properties in a desired way A good example is provided by the 3×3×3 “banana”-data set for the finger kinematics (Fig. 8b). Here, the metric distances of the given data points hardly would reveal their origin from a 3x3x3 sampling grid. Yet, the interpolation by a 3×3×3 PSOM (Fig. 8e) is very accurate and fully recovers the true topology of the data set.

If the underlying "true topology" of the data set is unknown, the choice of some grid $\mathbf{A}$ can only be considered a good guess. Therefore, the PSOM-approach is mainly suited for situations where the training data points are known to have arisen from some process with an underlying grid-like-sampling topology. This is frequently fulfilled in robotics, and in these cases then the proper labeling of the data points by the knot values $\mathbf{a}$ is either known or can be reconstructed, e.g. by the help of a standard SOM.

6 Conclusion

The PSOM algorithm can be characterized by *learning from very few examples* and dynamical assignment of input and output space (*associative completion*). Its strength is to pick up the essential curvature information of the topologically ordered training data set by constructing a smooth manifold. The kinematics of a 3 DOF robot finger can serve as a good example, see the "banana" reconstruction of Fig. 7 in Fig. 8e using the coarse data set Fig. 8b.

We presented two extensions to the PSOM algorithm aimed at improving the mapping accuracy and the computational efficiency with larger training sets.

(i) The proposed "local PSOM" algorithm constructs the constant sized PSOM on a dynamically determined sub-grid and keeps the computational effort constant when increasing the number of training points. Our result suggest an excellent cost–benefit relation when using more than four knots per axes.

(ii) An alternative to improve the mapping accuracy is the use of the "Chebyshev spaced PSOM" exploiting the superior approximation capabilities of the Chebyshev polynomials for the design of the internal basis functions. This imposes no extra effort but offers a significant precision advantage when using four or more knots per axes.

7 Appendix A: Basis Functions

A favorable choice for $H(\mathbf{a}, \mathbf{s})$ is the multidimensional extension of the well known Lagrange polynomial. The Lagrange formula describes the unique polynomial of degree $n-1$ passing through n support points $(x_i, y_i), i \in \{1, \ldots, n\}$

$$y(x) = l_1(x)y_1 + l_2(x)y_2 + \cdots + l_n(x)y_n = \sum_{k=1}^{n} l_k(x)y_k \qquad (4)$$

where the Lagrange factors $l_i(x)$ are determined by

$$l_i(x) = \prod_{j=1,\, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}. \qquad (5)$$

We extend now the one-to-one-dimensional ($x \mapsto y$) Lagrange interpolation formula (4) to m-to-n dimensional mapping S to X , using a knot set of supporting vectors $\mathbf{w_a}$ laying on a chosen rectangular hyper-grid $\mathbf{A}$ (before y_k). To clarify this we write it here once in long form: x becomes $\mathbf{s} = ({}^1s, {}^2s, \ldots, {}^ms)^T \in S \subset \mathbb{R}^m$ (the left upper index indicates here the vector component number); the support point coordinate x_i becomes a vector $\mathbf{a_i} = ({}^1a_{i_1}, {}^2a_{i_2}, \ldots, {}^ma_{i_m})^T \in \mathbf{A} \in S$. The knot set

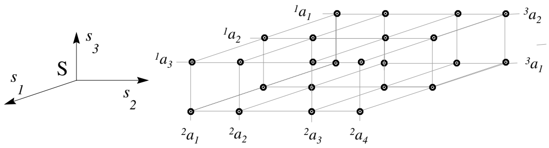

Figure 10: Example of a $m = 3$ dimensional mapping manifold S with a $3 \times 4 \times 2$ knot set $\mathbf{A} \subset S$ in orthonormal projection. Note the rectangular structure and the non equidistant spacing of the ${}^2a_i$.

$\mathbf{A} = \{{}^1a_1, \ldots, {}^1a_{n_1}\} \times \{{}^2a_1, \ldots, {}^2a_{n_2}\} \times \cdots \times \{{}^ma_1, \ldots, {}^ma_{n_m}\}$ containing obviously $n_1 \times n_2 \times \cdots \times n_m$ knots. Figure 10 illustrates an example with $n_1 = 3$, $n_2 = 4$, and $n_3 = 2$.

If we identify the knot $\mathbf{w_a} \in \mathbb{R}^d$ by its knot index numbers $i_\nu \in \{1, 2, \ldots, n_\nu\}$, $\mathbf{w_a} = \mathbf{w}_{i_1 i_2 \cdots i_m}$, we can expand equation (2) to

$$\mathbf{w}(\mathbf{s}) = \sum_{\mathbf{a} \in \mathbf{A}} \mathbf{w_a}\, H(\mathbf{a}, \mathbf{s}) = \sum_{\mathbf{a} \in \mathbf{A}} \mathbf{w}_{i_1 i_2 \cdots i_m} \cdot l_{i_1}({}^1s) \cdot l_{i_2}({}^2s) \cdots l_{i_m}({}^ms) \tag{6}$$

with

$$l_{i_\nu}({}^\nu s) = \prod_{j=1,\, j \neq i_\nu}^{n_\nu} \frac{{}^\nu s - {}^\nu a_j}{{}^\nu a_{i_\nu} - {}^\nu a_j} \tag{7}$$

The sum over $\mathbf{a}$ expands to the set of all possible indexes tuples $1 \le i_1 \le n_1$, $1, \le i_2 \le n_2$, $\cdots$, $1 \le i_m \le n_m$. The PSOM algorithm is invariant to rescaling of the S axes.

References

[1] P.J. Davis. *Interpolation and Approximation*. Dover Pub., New York, 1975.

[2] F. Girosi and T. Poggio. Networks and the best approximation property. *Biol. Cybern.*, 63(3):169–176, 1990.

[3] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer Series in Information Sciences 8. Springer, Heidelberg, 1984.

[4] R. Menzel, K. Woelfl, and F. Pfeiffer. The developement of a hydraulic hand. In *Proc. of the 2nd Conf. on Mechanotronics and Robotics*, pages 225–238, Sept 1993.

[5] John Moody and Christian Darken. Learning with localized receptive fields. In *Proc. Connectionist Models Summer School*, pages 133–143. Morgan Kaufman Publishers, San Mateo, CA, 1988.

[6] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge Univ. Press, 1988.

[7] Helge Ritter. Parametrized self-organizing maps. In S. Gielen and B. Kappen, editors, *ICANN'93-Proceedings, Amsterdam*, pages 568–575. Springer Verlag, Berlin, 1993.

[8] Helge Ritter. Parametrized self-organizing maps for vision learning tasks. In *ICANN'94-Proceedings, Italy*, pages 803–810, 1994.

[9] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neural Computation and Self-organizing Maps*. Addison Wesley, 1992.

[10] Jörg Walter and Helge Ritter. Investment learning with hierarchical PSOM. In *NIPS*95*, page (in press). MIT Press, 1995.

[11] Jörg Walter and Helge Ritter. The PSOM algorithm. *(in preparation)*, 1995.

[12] Jörg Walter and Helge Ritter. Rapid learning with parametrized self-organizing maps. *Neurocomputing, Special Issue*, (in press), 1995.