Hierarchical GTM: constructing localized non-linear projection manifolds in a principled way

Peter Tiňo Ian Nabney

Abstract

It has been argued that a single two-dimensional visualization plot may not be sufficient to capture all of the interesting aspects of complex data sets, and therefore a hierarchical visualization system is desirable. In this paper we extend an existing locally linear hierarchical visualization system PhiVis [1] in several directions: **(1)** We allow for *non-linear* projection manifolds. The basic building block is the Generative Topographic Mapping (GTM). **(2)** We introduce a general formulation of hierarchical probabilistic models consisting of local probabilistic models organized in a hierarchical tree. General training equations are derived, regardless of the position of the model in the tree. **(3)** Using tools from differential geometry we derive expressions for local directional curvatures of the projection manifold.

Like PhiVis, our system is statistically principled and is built interactively in a top-down fashion using the EM algorithm. It enables the user to interactively highlight those data in the ancestor visualization plots which are captured by a child model. We also incorporate into our system a hierarchical, locally selective representation of magnification factors and directional curvatures of the projection manifolds. Such information is important for further refinement of the hierarchical visualization plot, as well as for controlling the amount of regularization imposed on the local models. We demonstrate the principle of the approach on a toy data set and apply our system to two more complex 12- and 18-dimensional data sets.

Keywords

Hierarchical probabilistic model, Generative Topographic Mapping, data visualization, EM algorithm, density estimation, directional curvature.

I. Introduction

MOST data visualization algorithms project the data onto a two-dimensional visualization space. However, a single two-dimensional projection, even if it is

This work was supported by the BBSRC grant BIO/12093 and Pfizer Research. The authors are with the Neural Computation Research Group, Aston University, Aston Triangle, Birmingham. B4 7ET, UK. Corresponding author: PT, p.tino@aston.ac.uk.

non-linear, may not be sufficient to capture all of the interesting aspects of the data. This motivated Bishop and Tipping [1] to develop a hierarchical model involving multiple linear two-dimensional visualization spaces. The intuition behind their approach is that the lack of flexibility of individual models can be compensated for by the overall flexibility of the complete hierarchy. However, there are situations where using a hierarchy of non-linear models can lead to more natural and parsimonious data representations. Consider, for example, a set of points close to the two-dimensional manifold shown in figure 1. The set could be covered by a large number of linear two-dimensional sheets, but in this case, a collection of four simple non-linear "humps" is a more natural alternative. Of course, as discussed in this paper, once we allow for non-linear local projections, we need an effective mechanism to control the "amount of non-linearity" in the projection manifolds. To this end, we visualize in a hierarchical and interactive way the local magnification factors and directional curvatures of the projection manifolds.

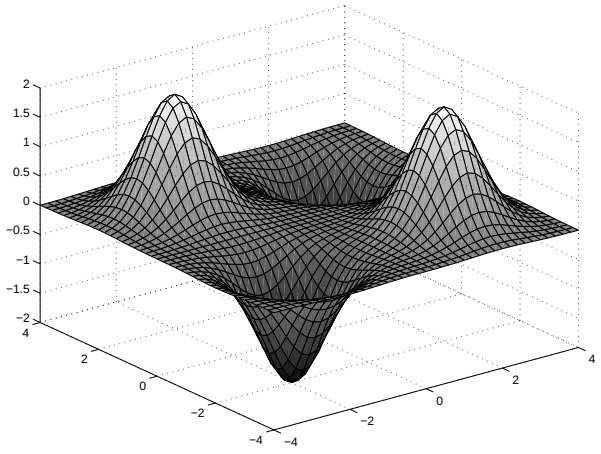

Fig. 1. A two-dimensional manifold in three-dimensional Euclidean space.

When investigating a data set through low-dimensional projections in a hierarchical way, one usually first constructs a top-level plot and then concentrates on local regions of interest by recursively building the corresponding sub-projections. The sub-models are organized in a hierarchical tree and should ideally form a consistent probabilistic model of the data, as with the hierarchical locally linear model of Bishop and Tipping [1]. Here, we present a consistent probabilistic model of the data that performs *non-linear* local data projections.

The basic building block of our hierarchical model is the *Generative Topographic Map-*

ping (GTM) introduced by Bishop, Svensén and Williams [2]. It is a probabilistic reformulation of the self-organizing map (SOM) [3] and offers many advantages compared with the standard SOM [4], principally that it defines an explicit probability density model of the data. This enables us to apply the consistent and statistically principled framework used in [1] to formulate hierarchical non-linear visualization trees. Also, unlike SOMs, local GTMs form *smooth* two-dimensional manifolds on which quantities useful for monitoring the "amount of non-linearity", like magnification factors [5], or curvatures, can be computed analytically. Approaches to hierarchical data visualization that incorporated SOM [6] [7] [8] partitioned data in a "hard" fashion, while our approach permits "soft" partitioning in which, at any level of hierarchy, data points can effectively belong to more than one local model.

In a closely related field of data clustering Williams proposed a probabilistic mixture model that generates data in a hierarchical tree-structured manner [9]. The tree structure is inferenced from data using Markov Chain Monte Carlo (MCMC) methods. MCMC is used to sample from the posterior distribution over trees of variable size, given the data points and a prior over trees expressed as a Markovian model for numbers of nodes at different levels of the tree.

The paper has the following organization: In section II we give a general formulation of probabilistic models organized in hierarchical trees. Section III briefly introduces the basic building block of our visualization system – the Generative Topographic Mapping [2]. In section IV we derive equations for an EM algorithm that fits GTMs in the hierarchy to the data. Using tools of differential geometry, we show in section V how to compute local directional curvatures of the GTM projection manifold and briefly mention previous work on magnification factors. Section VI describes details of the implemented hierarchical visualization system and section VII presents the experiments on a toy three-dimensional data set and two more complex 12- and 18-dimensional data collections. The discussion in section VIII highlights the experimental findings and compares our system with the linear hierarchical visualization tool of Bishop and Tipping [1]. Section IX concludes the paper by summarizing the key contributions of this study.

II. Hierarchical probabilistic models

In this section, we give a general outline of hierarchical probabilistic models that consist of local probabilistic models $\mathcal{M}$ organized in hierarchical trees. Each model $\mathcal{M}$ defines a distribution $P(\mathbf{t}|\ \mathcal{M})$ on a data space $\mathcal{D}$, $\mathbf{t} \in \mathcal{D}$. First, we introduce notation that reflects the fact that hierarchical trees are special cases of graphs.

A. Hierarchical Trees

For the sake of simplicity, we illustrate the concepts on an example, generalization is straightforward.

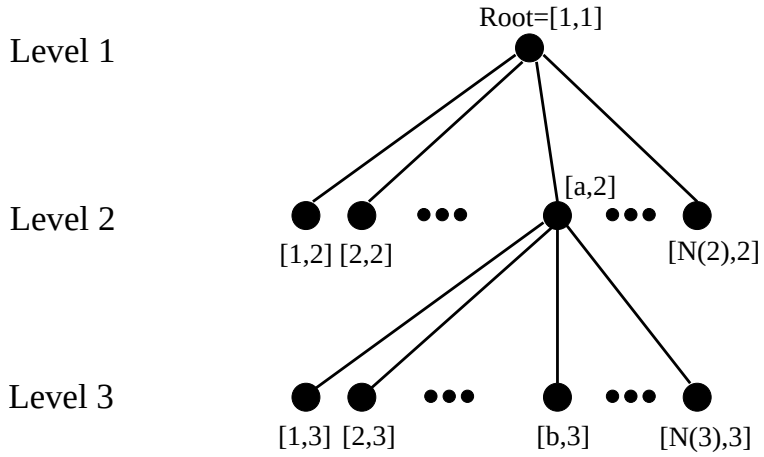

Fig. 2. An example of a hierarchical tree.

Consider a hierarchical tree $\mathcal{T}$ shown in figure 2. We introduce the following functions operating on nodes (probabilistic models on the data space $\mathcal{D}$) $\mathcal{M}$ of $\mathcal{T}$:

- **Parent**$(\mathcal{M})$ — the first-generation ancestor of $\mathcal{M}$

$Parent([a,2]) = Root,\ \ Parent([b,3]) = [a,2].$

- **Children**$(\mathcal{M})$ — the set of first-generation descendants of $\mathcal{M}$

$Children(Root) = \{[1,2],[2,2],...,[N(2),2]\},\ Children([a,2]) = \{[1,3],[2,3],...,[N(3),3]\}.$

- **Level**$(\mathcal{M})$ — level of $\mathcal{M}$ in $\mathcal{T}$

$Level(Root) = 1,\ \ Level([a,2]) = 2,\ \ Level([b,3]) = 3.$

- **Nodes**(ℓ) — the set of nodes at level ℓ,

$Nodes(\ell) = \{\mathcal{M}|\ Level(\mathcal{M}) = \ell\} = \bigcup_{\mathcal{M} \in Nodes(\ell-1)}\ Children(\mathcal{M})$

$Nodes(1) = \{Root\},\ \ Nodes(2) = \{[1,2],[2,2],...,[N(2),2]\}.$

- **Path**$(\mathcal{M})$ — N-tuple of nodes defining the path from $Root$ to $\mathcal{M}$, where $N = Level(\mathcal{M})$

$Path(Root) = (Root),\ \ Path([a,2]) = (Root,[a,2]),\ \ Path([b,3]) = (Root,[a,2],[b,3]),$

writing element-wise: $Path([b,3])_1 = Root,\ \ Path([b,3])_2 = [a,2],\ \ Path([b,3])_3 = [b,3].$

Leaves$(\mathcal{T})$ is the set of leaves of the tree $\mathcal{T}$, i.e. the set of nodes without children.

B. Model formulation

The hierarchical probabilistic model is obtained by interpreting the nodes of the hierarchical tree $\mathcal{T}$ as probabilistic models on the data space.

Each model $\mathcal{M}$ in the hierarchy, except for *Root*, has an associated parent-conditional mixture coefficient, or prior

$$\pi(\mathcal{M}|\ Parent(\mathcal{M})). \tag{1}$$

The priors are non-negative and satisfy the consistency condition:

- *for any model* $\mathcal{N}$ *having children,*

$$\sum_{\mathcal{M}\in Children(\mathcal{N})} \pi(\mathcal{M}|\ \mathcal{N}) = 1. \tag{2}$$

Unconditional priors for the models are recursively calculated as follows:

- prior for *Root* is unity

$$\pi(Root) = 1, \tag{3}$$

- and for all other models

$$\pi(\mathcal{M}) = \prod_{i=2}^{Level(\mathcal{M})} \pi(Path(\mathcal{M})_i|\ Path(\mathcal{M})_{i-1}). \tag{4}$$

Now, we are ready to write the distribution $P(\mathbf{t}|\ \mathcal{T})$ given by the hierarchical model; it is a mixture of models at the leaves of the tree $\mathcal{T}$,

$$P(\mathbf{t}|\ \mathcal{T}) = \sum_{\mathcal{M}\in Leaves(\mathcal{T})} \pi(\mathcal{M})\ P(\mathbf{t}|\ \mathcal{M}). \tag{5}$$

Models corresponding to internal (i.e. non-leaf) nodes of $\mathcal{T}$ play their role only in the process of creating the hierarchical model. Once the hierarchy is trained and mixture coefficients (4) are established, we need the internal models only if we wish to extend or retrain the hierarchical model structure in the future.

III. Generative Topographic Mapping

The Generative Topographic Mapping belongs to a family of latent space models that model a probability distribution in the (observable) data space by means of latent, or hidden variables. The latent space is used to visualize the data, and is usually a bounded subset of the two-dimensional Euclidean space, such as the unit square, or the (two-dimensional) interval $[-1,1] \times [-1,1]$.

Consider an L-dimensional latent space $\mathcal{H} \subset \Re^L$ of a GTM $\mathcal{M}$ and represent points in $\mathcal{H}$ as column vectors $\mathbf{x} = (x_1, x_2, ..., x_L)^T$. We discretize the latent space by introducing a regular array of latent space centres $\mathbf{x}_i^{\mathcal{M}} \in \mathcal{H}$, labelled by the index $i = 1, 2, ..., K_{\mathcal{M}}$. Latent space centres are analogous to the nodes of SOM.

Let the data space be the D-dimensional Euclidean space $\Re^D$. We define a non-linear transformation $f_{\mathcal{M}} : \mathcal{H} \to \Re^D$ from the latent space to the data space using a radial basis function network (see e.g. [10]). To this end, we cover the latent space with a set of $M_{\mathcal{M}} - 1$ fixed non-linear basis functions $\phi_j : \mathcal{H} \to \Re$, $j = 1, 2, ..., M_{\mathcal{M}} - 1$, which form a non-orthogonal basis set. In this paper, as usual in the GTM literature, we choose to work with spherical Gaussian functions of the same width σ, although other choices are possible and require simple modifications. The centres of the Gaussian basis functions ϕ_j are positioned in the latent space on a regular grid. This is because the basis functions should model the latent space density (see [10]) which is defined to be uniform. To account for the bias term, we introduce an additional constant basis function $\phi_{M_{\mathcal{M}}}(\mathbf{x}) = 1$, for all $\mathbf{x} \in \mathcal{H}$. Given a point $\mathbf{x} \in \mathcal{H}$, the values given by the basis functions at $\mathbf{x}$ are summarized by a column vector

$$\phi_{\mathcal{M}}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_{M_{\mathcal{M}}}(\mathbf{x}))^T, \tag{6}$$

and the image of $\mathbf{x}$ under the map $f_{\mathcal{M}}$ is computed as

$$f_{\mathcal{M}}(\mathbf{x}) = \mathbf{W}_{\mathcal{M}} \ \phi_{\mathcal{M}}(\mathbf{x}), \tag{7}$$

where $\mathbf{W}_{\mathcal{M}}$ is a $D \times M_{\mathcal{M}}$ matrix of weights.

GTM creates a generative probabilistic model in the data space by placing a radially-symmetric Gaussian with zero mean and inverse variance $\beta_{\mathcal{M}}$ around images, under $f_{\mathcal{M}}$,

of the latent space centres $\mathbf{x}_i^{\mathcal{M}} \in \mathcal{H}$, $i = 1, 2, ..., K_{\mathcal{M}}$:

$$P(\mathbf{t}|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}}) = \left(\frac{\beta_{\mathcal{M}}}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta_{\mathcal{M}}}{2}\|f(\mathbf{x}_i^{\mathcal{M}}) - \mathbf{t}\|^2\right\}. \tag{8}$$

Defining a uniform prior over $\mathbf{x}_i^{\mathcal{M}}$, the density model in the data space provided by the GTM $\mathcal{M}$ is then

$$P(\mathbf{t}|\ \mathcal{M}) = \frac{1}{K_{\mathcal{M}}} \sum_{i=1}^{K_{\mathcal{M}}} P(\mathbf{t}|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}}). \tag{9}$$

Given a data set $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$ of i.i.d. points in the data space, the adjustable parameters $\mathbf{W}_{\mathcal{M}}$ and $\beta_{\mathcal{M}}$ of the model $\mathcal{M}$ can be fitted to the data by maximum likelihood. The log likelihood function is given by

$$\mathcal{L}(\mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}}) = \sum_{n=1}^{N} \ln P(\mathbf{t}_n|\ \mathcal{M}). \tag{10}$$

The log likelihood can be maximized using a gradient-based procedure, or the expectation-maximization (EM) algorithm [11]. A derivation of the EM algorithm for GTM can be found in [2].

For the purpose of data visualization, we use Bayes' theorem to invert the transformation $f_{\mathcal{M}}$ from the latent space $\mathcal{H}$ to the data space $\mathcal{D}$. Since we choose to work with a prior distribution on $\mathcal{H}$ that effectively discretizes the latent space into the grid $\mathbf{x}_i^{\mathcal{M}}$, $i = 1, 2, ..., K_{\mathcal{M}}$, the posterior distribution on $\mathcal{H}$, given a data point $\mathbf{t}_n \in \mathcal{D}$, is a sum of delta functions centered at $\mathbf{x}_i^{\mathcal{M}}$, with coefficients given by the responsibilities

$$R_{i,n}^{\mathcal{M}} = \frac{P(\mathbf{t}_n|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}})}{\sum_{j=1}^{K_{\mathcal{M}}} P(\mathbf{t}_n|\ \mathbf{x}_j^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}})}. \tag{11}$$

The responsibility $R_{i,n}^{\mathcal{M}}$ is the posterior probability that the Gaussian $P(\mathbf{t}_n|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}})$ generated the point $\mathbf{t}_n$ in the data space. When used for data visualization, GTM $\mathcal{M}$ projects points $\mathbf{t}_n$ from the data space into the low-dimensional latent space $\mathcal{H}$. The latent space representation of the point $\mathbf{t}_n$ is taken to be the mean

$$\sum_{i-1}^{K_{\mathcal{M}}} R_{i,n}^{\mathcal{M}}\ \mathbf{x}_i^{\mathcal{M}}, \tag{12}$$

or the mode

$$\mathbf{x}_{i^*}, \quad i^* = \arg\max_{\{i\}} R_{i,n}^{\mathcal{M}} \tag{13}$$

of the posterior distribution on $\mathcal{H}$.

The $f_{\mathcal{M}}$–image of the latent space $\mathcal{H}$,

$$\Omega = f_{\mathcal{M}}(\mathcal{H}) = \{f_{\mathcal{M}}(\mathbf{x}) \in \Re^D | \ \mathbf{x} \in \mathcal{H}\}, \tag{14}$$

forms an L-dimensional manifold in the data space. We refer to the manifold Ω as the *projection manifold* of GTM $\mathcal{M}$.

IV. Training the hierarchy of GTMs

Training of a hierarchy of GTMs proceeds in a recursive fashion. First, a root GTM is trained and used to visualize the data. Then the user identifies interesting regions on the visualization plot that they would like to model in a greater detail. These "regions of interest" are then transformed into the data space and form the basis for building a collection of new, child GTMs. After seeing the lower level visualization plots, the user may decide to proceed further and model in a greater detail some portions of the lower level plots, etc.

In the following, we assume that we have already trained a hierarchy of GTMs up to level ℓ of a hierarchical tree $\mathcal{T}$. The purpose of this section is to formulate the EM algorithm that fits child GTMs $\mathcal{M}$, of models $\mathcal{N}$ at level ℓ, to the data set $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$. Child GTMs of models at level ℓ are GTMs at level $\ell + 1$. The current stage of the hierarchical GTM construction is shown in figure 3.

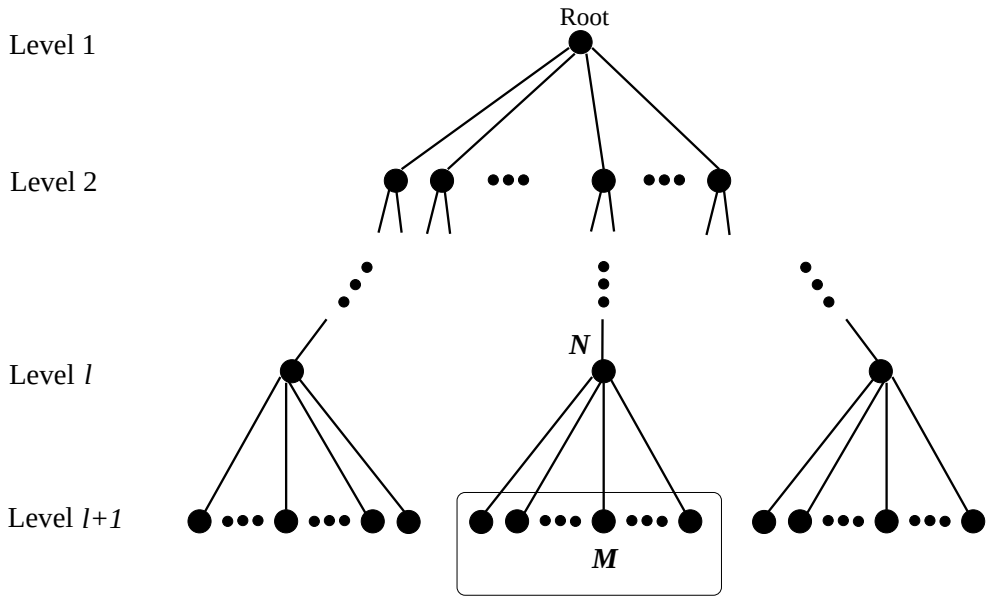

Fig. 3. A stage in the hierarchical GTM construction. All GTMs up to level ℓ have been built. Now, child GTMs $\mathcal{M}$ at level $\ell + 1$ of the parent GTMs $\mathcal{N}$ at level ℓ are being constructed.

A. The EM algorithm

Given the training data $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$, the likelihood function of the hierarchy $\mathcal{T}$ of GTMs is

$$\mathcal{L} = \sum_{n=1}^{N} \ln P(\mathbf{t}_n |\ \mathcal{T}), \tag{15}$$

where $P(\mathbf{t} |\ \mathcal{T})$ is given by (5).

We fit children of the parent GTMs $\mathcal{N}$ at level ℓ to the training set by maximizing the likelihood function $\mathcal{L}$. At the current stage, the children of the models $\mathcal{N}$ are leaves of the hierarchical tree $\mathcal{T}$, and so the distribution $P(\mathbf{t} |\ \mathcal{T})$ given by the hierarchical model can be rewritten as

$$P(\mathbf{t} |\ \mathcal{T}) = \sum_{\mathcal{M} \in Leaves(\mathcal{T})} \pi(\mathcal{M})\ P(\mathbf{t} |\ \mathcal{M}) \ = \ Q_{\mathcal{T} \setminus Nodes(\ell+1)}(\mathbf{t}) \ + \ Q_{Nodes(\ell+1)}(\mathbf{t}), \tag{16}$$

where

$$Q_{\mathcal{T} \setminus Nodes(\ell+1)}(\mathbf{t}) = \sum_{\mathcal{M} \in Leaves(\mathcal{T}) \setminus Nodes(\ell+1)} \pi(\mathcal{M})\ P(\mathbf{t} |\ \mathcal{M}) \tag{17}$$

and

$$Q_{Nodes(\ell+1)}(\mathbf{t}) = \sum_{\mathcal{M} \in Nodes(\ell+1)} \pi(\mathcal{M})\ P(\mathbf{t} |\ \mathcal{M}). \tag{18}$$

Since all GTMs in the hierarchy, except for the recently added models in $Nodes(\ell+1)$, are fixed, the likelihood function $\mathcal{L}$ is maximized by maximizing the *restricted likelihood function* confined only to the GTMs at level $\ell + 1$,

$$\mathcal{L}^{(\ell+1)} = \sum_{n=1}^{N} \ln Q_{Nodes(\ell+1)}(\mathbf{t}_n). \tag{19}$$

From (4), the mixture coefficients $\pi(\mathcal{M})$ of a GTM $\mathcal{M}$ at level $\ell + 1$ are given by,

$$\pi(\mathcal{M}) = \pi(\mathcal{M} |\ Parent(\mathcal{M}))\ \pi(Parent(\mathcal{M})), \tag{20}$$

and so (18) becomes

$$Q_{Nodes(\ell+1)}(\mathbf{t}) = \sum_{\mathcal{M} \in Nodes(\ell+1)} \pi(\mathcal{M} |\ Parent(\mathcal{M}))\ \pi(Parent(\mathcal{M}))\ P(\mathbf{t} |\ \mathcal{M}), \tag{21}$$

giving the restricted likelihood function

$$\mathcal{L}^{(\ell+1)} = \sum_{n=1}^{N} \ln \left\{ \sum_{\mathcal{M} \in Nodes(\ell+1)} \pi(\mathcal{M} |\ Parent(\mathcal{M}))\ \pi(Parent(\mathcal{M}))\ P(\mathbf{t}_n |\ \mathcal{M}) \right\}. \tag{22}$$

If we knew, before adding children to GTMs at level ℓ, which GTM at level ℓ generated which point in the data set $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$, we would be able to rewrite (22) as

$$\mathcal{L}^{(\ell+1)} = \sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} \nu_{n,\mathcal{N}} \ \ln \left\{ \sum_{\mathcal{M} \in Children(\mathcal{N})} \pi(\mathcal{M}|\ \mathcal{N})\ \pi(\mathcal{N})\ P(\mathbf{t}_n|\ \mathcal{M}) \right\}, \qquad (23)$$

where the assignment variables $\nu_{n,\mathcal{N}}$ are 1, if GTM $\mathcal{N}$ was responsible for generating the point $\mathbf{t}_n$, and 0 otherwise.

In reality, we do not know the values of the assignments $\nu_{n,\mathcal{N}}$, but we do know the posterior probabilities $P(\mathcal{N}|\ \mathbf{t}_n)$ that GTM $\mathcal{N}$ generated $\mathbf{t}_n$. We also refer to these posteriors as the responsibilities of $\mathcal{N}$ for generating $\mathbf{t}_n$. These were calculated in the previous stage of the training and are now fixed. We will later show how to calculate the posteriors $P(\mathcal{M}|\ \mathbf{t}_n)$ for models $\mathcal{M}$ at level $\ell + 1$.

Taking the expectation of (23), we arrive at expected restricted likelihood function for models at level $\ell + 1$,

$$\left\langle \mathcal{L}^{(\ell+1)} \right\rangle = \sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} P(\mathcal{N}|\ \mathbf{t}_n)\ \ln \left\{ \pi(\mathcal{N}) \sum_{\mathcal{M} \in Children(\mathcal{N})} \pi(\mathcal{M}|\ \mathcal{N})\ P(\mathbf{t}_n|\ \mathcal{M}) \right\}. \quad (24)$$

Now, imagine that given information that a parent model $\mathcal{N}$ was indeed responsible for generating a point $\mathbf{t}_n$, we knew which of its children $\mathcal{M}$ generated $\mathbf{t}_n$. We represent this (hypothetical) situation by assignment variables $\nu_{n,\mathcal{M}|\mathcal{N}}$. In reality, we are only able to compute (parent-conditional) responsibilities $P(\mathcal{M}|\ \mathcal{N}, \mathbf{t}_n)$.

Our probabilistic models are GTMs that model probability distribution in the data space in terms of hidden variables (see section III). Suppose for a moment that we knew which latent space centre $\mathbf{x}_i^{\mathcal{M}} \in \mathcal{H}$, $i = 1, 2, ..., K_{\mathcal{M}}$, of the GTM $\mathcal{M}$ corresponded to the Gaussian that generated $\mathbf{t}_n$ (eq. (8)). Again, we represent this hypothetical situation by assignment variables $z_{n,i}^{\mathcal{M}}$. Since the latent variables $\mathbf{x}_i^{\mathcal{M}}$ are hidden, we only have the responsibilities $R_{i,n}^{\mathcal{M}}$ given by eq. (11).

To recapitulate, we have two types of hidden variables:

- the assignment variables $\nu_{n,\mathcal{M}|\mathcal{N}}$ that group children $\mathcal{M}$ of the GTM $\mathcal{N}$ in a mixture model
- the assignment variables $z_{n,i}^{\mathcal{M}}$ formulating GTM $\mathcal{M}$ as a constrained[1] mixture of Gaus-

[1] GTM is considered a constrained mixture of Gaussians, because the means of the Gaussians (8) are constrained to lie on the $f_{\mathcal{M}}$–image of the latent space (i.e. on the projection manifold of the GTM $\mathcal{M}$), which is a low-dimensional manifold in the data space.

sians.

If we knew the values of the assignment variables, the expected restricted likelihood function (24) could be written as the complete-data likelihood restricted to models at level $\ell + 1$,

$$\mathcal{L}_C^{(\ell+1)} = \sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} P(\mathcal{N}|\ \mathbf{t}_n) \sum_{\mathcal{M} \in Children(\mathcal{N})} \nu_{n,\mathcal{M}|\mathcal{N}} \sum_{i=1}^{K_{\mathcal{M}}} z_{n,i}^{\mathcal{M}}\ \ln \left\{ \pi(\mathcal{N})\ \pi(\mathcal{M}|\ \mathcal{N})\ P(\mathbf{t}_n, \mathbf{x}_i^{\mathcal{M}}) \right\}. \tag{25}$$

Taking expectation over both types of hidden variables we arrive at the expected restricted complete-data likelihood

$$\left\langle \mathcal{L}_C^{(\ell+1)} \right\rangle = \sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} P(\mathcal{N}|\ \mathbf{t}_n) \sum_{\mathcal{M} \in Children(\mathcal{N})} P(\mathcal{M}|\ \mathcal{N}, \mathbf{t}_n) \sum_{i=1}^{K_{\mathcal{M}}} R_{i,n}^{\mathcal{M}}\ \ln \left\{ \pi(\mathcal{N})\ \pi(\mathcal{M}|\ \mathcal{N})\ P(\mathbf{t}_n, \mathbf{x}_i^{\mathcal{M}}) \right\}. \tag{26}$$

Since

$$P(\mathbf{t}_n, \mathbf{x}_i^{\mathcal{M}}) = P(\mathbf{t}|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}})\ P(\mathbf{x}_i^{\mathcal{M}}),$$

where $P(\mathbf{t}|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}})$ is given by (8), and $P(\mathbf{x}_i^{\mathcal{M}})$ is a uniform prior

$$P(\mathbf{x}_i^{\mathcal{M}}) = \frac{1}{K_{\mathcal{M}}},$$

to maximize $\left\langle \mathcal{L}_C^{(\ell+1)} \right\rangle$, we need to consider only two terms:

$$\sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} P(\mathcal{N}|\ \mathbf{t}_n) \sum_{\mathcal{M} \in Children(\mathcal{N})} P(\mathcal{M}|\ \mathcal{N}, \mathbf{t}_n)\ \ln \pi(\mathcal{M}|\ \mathcal{N}) \tag{27}$$

and

$$\sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} P(\mathcal{N}|\ \mathbf{t}_n) \sum_{\mathcal{M} \in Children(\mathcal{N})} P(\mathcal{M}|\ \mathcal{N}, \mathbf{t}_n) \sum_{i=1}^{K_{\mathcal{M}}} R_{i,n}^{\mathcal{M}}\ \ln P(\mathbf{t}_n|\ \mathbf{x}_i^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}}). \tag{28}$$

The remaining term

$$\sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(\ell)} P(\mathcal{N}|\ \mathbf{t}_n)\ \ln \pi(\mathcal{N})$$

is constant with respect to the adjustable parameters of GTMs at level $\ell + 1$.

The M-step of the EM algorithm involves maximizing (27) with respect to the parent-conditional mixture coefficients $\pi(\mathcal{M}|\,\mathcal{N})$ and maximizing (28) with respect to the GTMs' parameters $\mathbf{W}_{\mathcal{M}}$ and $\beta_{\mathcal{M}}$.

The maximization of (27) with respect to $\pi(\mathcal{M}|\ \mathcal{N})$ must take account of the constraint

$$\sum_{\mathcal{M} \in Children(\mathcal{N})} \pi(\mathcal{M}|\ \mathcal{N}) = 1.$$

This can be achieved by introducing a Lagrange multiplier $\lambda_{\mathcal{N}}$ (see [1] [10]) and maximizing

$$\sum_{n=1}^{N}\ P(\mathcal{N}|\ \mathbf{t}_n) \sum_{\mathcal{M} \in Children(\mathcal{N})} P(\mathcal{M}|\ \mathcal{N}, \mathbf{t}_n)\ \ln \pi(\mathcal{M}|\ \mathcal{N})\ +\ \lambda_{\mathcal{N}} \left(\sum_{\mathcal{M} \in Children(\mathcal{N})} \pi(\mathcal{M}|\ \mathcal{N}) \right).$$

After a straightforward calculation, we obtain

$$\pi(\mathcal{M}|Parent(\mathcal{M})) = \frac{\sum_{n=1}^{N} P(\mathcal{M}|\ \mathbf{t}_n)}{\sum_{n=1}^{N} P(Parent(\mathcal{M})|\ \mathbf{t}_n)}, \tag{29}$$

where

$$P(\mathcal{M}|\ \mathbf{t}_n) = P(\mathcal{M}|\ Parent(\mathcal{M}), \mathbf{t}_n)\ P(Parent(\mathcal{M})|\ \mathbf{t}_n). \tag{30}$$

Maximizing (28) with respect to $\mathbf{W}_{\mathcal{M}}$, using (6), (7) and (8), we obtain

$$\sum_{n=1}^{N} P(\mathcal{M}|\ \mathbf{t}_n)\ \sum_{i=1}^{K_{\mathcal{M}}} R_{i,n}^{\mathcal{M}}\ \left(\mathbf{W}_{\mathcal{M}}\ \phi_{\mathcal{M}}(\mathbf{x}_i^{\mathcal{M}}) - \mathbf{t}_n \right)\ \phi_{\mathcal{M}}^T(\mathbf{x}_i^{\mathcal{M}}) = 0. \tag{31}$$

The responsibilities $R_{i,n}^{\mathcal{M}}$ are calculated with the current ("old") weight and inverse variance parameters of the child GTMs $\mathcal{M}$.

Written in matrix notation, we have to solve

$$(\mathbf{\Phi}_{\mathcal{M}}^T\ \mathbf{B}_{\mathcal{M}}\ \mathbf{\Phi}_{\mathcal{M}})\ \mathbf{W}_{\mathcal{M}}^T = \mathbf{\Phi}_{\mathcal{M}}^T\ \mathbf{R}_{\mathcal{M}}\ \mathbf{T} \tag{32}$$

for $\mathbf{W}_{\mathcal{M}}$.

The above system of linear equations involves the following matrices:

- $\mathbf{\Phi}_{\mathcal{M}}$ is a $K_{\mathcal{M}} \times M_{\mathcal{M}}$ matrix with elements (see eq. (6))

$$(\mathbf{\Phi}_{\mathcal{M}})_{ij} = \phi_j(\mathbf{x}_i^{\mathcal{M}}), \tag{33}$$

- $\mathbf{T}$ is a $N \times D$ matrix storing the data points $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N$ as rows,
- $\mathbf{R}_{\mathcal{M}}$ is a $K_{\mathcal{M}} \times N$ matrix containing, for each latent space centre $\mathbf{x}_i^{\mathcal{M}}$, and each data point $\mathbf{t}_n$, *scaled responsibilities*

$$(\mathbf{R}_{\mathcal{M}})_{in} = P(\mathcal{M}|\ \mathbf{t}_n)\ R_{i,n}^{\mathcal{M}} \tag{34}$$

computed using (30) and (11),

• $\mathbf{B}_{\mathcal{M}}$ is a $K_{\mathcal{M}} \times K_{\mathcal{M}}$ diagonal matrix with diagonal elements corresponding to responsibilities of latent space centres for the whole data sample $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$,

$$(\mathbf{B}_{\mathcal{M}})_{ii} = \sum_{n=1}^{N} P(\mathcal{M}|\ \mathbf{t}_n)\ R_{i,n}^{\mathcal{M}}. \tag{35}$$

The GTM mapping $f_{\mathcal{M}}$ can be regularized by adding a regularization term to the likelihood (10). Bishop, Svensén and Williams [4] suggest to use a quadratic regularizer of the form

$$\frac{1}{2}\ \alpha_{\mathcal{M}}\ \|\mathrm{vec}(\mathbf{W}_{\mathcal{M}}^T)\|^2, \tag{36}$$

where $\mathrm{vec}(\mathbf{W}_{\mathcal{M}}^T)$ is a column vector consisting of the concatenation of the successive columns of the weight matrix $\mathbf{W}_{\mathcal{M}}$, and $\alpha_{\mathcal{M}}$ is the regularization coefficient. Inclusion of the regularizer (36) modifies eq. (32) to

$$\left[\mathbf{\Phi}_{\mathcal{M}}^T\ \mathbf{B}_{\mathcal{M}}\ \mathbf{\Phi}_{\mathcal{M}}\ +\ \frac{\alpha_{\mathcal{M}}}{\beta_{\mathcal{M}}}\mathbf{I}\right]\ \mathbf{W}_{\mathcal{M}}^T = \mathbf{\Phi}_{\mathcal{M}}^T\ \mathbf{R}_{\mathcal{M}}\ \mathbf{T} \tag{37}$$

where $\mathbf{I}$ is the $M_{\mathcal{M}} \times M_{\mathcal{M}}$ identity matrix.

Finally, maximizing (28) with respect to $\beta_{\mathcal{M}}$ leads to the re-estimation formula (see (7), (11), and (30))

$$\frac{1}{\beta_{\mathcal{M}}} = \frac{\sum_{n=1}^{N} P(\mathcal{M}|\ \mathbf{t}_n)\ \sum_{i=1}^{K_{\mathcal{M}}}\ R_{i,n}^{\mathcal{M}}\ \ \|\mathbf{W}_{\mathcal{M}}\ \phi(\mathbf{x}_i^{\mathcal{M}}) - \mathbf{t}_n\|^2}{D\ \sum_{n=1}^{N} P(\mathcal{M}|\ \mathbf{t}_n)}, \tag{38}$$

where $\mathbf{W}_{\mathcal{M}}$ is the "new" weight matrix computed by solving (32) in the last step.

In the E-step of the EM algorithm we estimate the latent space responsibilities $R_{i,n}^{\mathcal{M}}$ within individual GTMs (eq. (11)), model responsibilities $P(\mathcal{M}|\ \mathbf{t}_n)$ (eq. (30)), and parent-conditional model responsibilities

$$P(\mathcal{M}|\ Parent(\mathcal{M}), \mathbf{t}_n) = \frac{\pi(\mathcal{M}|\ Parent(\mathcal{M}))\ P(\mathbf{t}_n|\ \mathcal{M})}{\sum_{\mathcal{N} \in [\mathcal{M}]} \pi(\mathcal{N}|\ Parent(\mathcal{M}))\ P(\mathbf{t}_n|\ \mathcal{N})}, \tag{39}$$

where

$$[\mathcal{M}] = Children(Parent(\mathcal{M})). \tag{40}$$

B. Summary of the EM algorithm

Hierarchical GTM is trained using EM to maximize its likelihood with respect to the data sample $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$. The hierarchy is trained in a top-down fashion, starting with the root model, then continuing with its children, then with children of the children,

etc. At each stage of hierarchical GTM construction, the EM algorithm alternates between the E- and M-steps until convergence is satisfactory (typically after 10–20 iterations). To avoid numerical problems arising from multiplication of small probabilities and to speed up the training process, the GTMs on lover levels are trained only on data points for which the parent model has responsibility greater than some pre-set threshold ϵ. In our experiments $\epsilon = 10^{-5}$.

To make expressions for training individual models consistent throughout the hierarchy, we introduce a virtual model $Parent(Root)$ by postulating

$$\begin{aligned} \pi(Root|\ Parent(Root)) &= 1, \\ Children(Parent(Root)) &= \{Root\}, \\ P(Parent(Root)|\ \mathbf{t}_n) &= 1. \end{aligned} \tag{41}$$

We also set

$$P(Root|\ \mathbf{t}_n) = 1. \tag{42}$$

B.1 E-step

In the E-step, we estimate posterior over all hidden variables, using the "old" values of GTM parameters.

- Given a data point $\mathbf{t}_n \in \Re^D$, (39) is used to compute the model responsibilities corresponding to the competition among models belonging to the same parent.
- The unconditional (on parents) model responsibilities are recursively determined by (30).
- Responsibilities of the latent space centres $\mathbf{x}_i^{\mathcal{M}}$, $i = 1, 2, ..., K_{\mathcal{M}}$, corresponding to the competition among the latent space centres in each model $\mathcal{M}$ are calculated using (11).

B.2 M-step

In the M-step, we estimate the parameters using the posterior over hidden variables computed in the E-step.

- Parent-conditional mixture coefficients are determined by (29).
- Weight matrices $\mathbf{W}_{\mathcal{M}}$ are calculated by solving (32) using standard inversion techniques based on singular value decomposition [12] to allow for possible ill-conditioning.
- The inverse variances are re-estimated using (38).

C. Parameter initialization

Having trained GTMs up to level ℓ of the hierarchical tree $\mathcal{T}$, we pick a parent model $\mathcal{N}$ at level ℓ and, based on its visualization plot, we select regions of interest for child GTMs $\mathcal{M}$ at level $\ell + 1$. The regions of interest are selected as follows: The user first selects points $\mathbf{c}_i \in \mathcal{H}$, $i = 1, 2, ..., A$, in the latent space that correspond to "centres" of the subregions they are interested in. The points $\mathbf{c}_i$ are then transformed via the map $f_{\mathcal{N}}$, defined by the parent GTM, to the data space (eq. (7))

$$f_{\mathcal{N}}(\mathbf{c}_i) = \mathbf{W}_{\mathcal{N}} \ \phi_{\mathcal{N}}(\mathbf{c}_i).$$

The regions of interest are given by the Voronoi compartments [13] in the data space corresponding to the points $f_{\mathcal{N}}(\mathbf{c}_i)$, $i = 1, 2, ..., A$:

$$V_i = \left\{ \mathbf{t} \in \Re^D | \ d\left(\mathbf{t}, f_{\mathcal{N}}(\mathbf{c}_i)\right) = \min_j \ d\left(\mathbf{t}, f_{\mathcal{N}}(\mathbf{c}_j)\right) \right\}, \tag{43}$$

where $d(\cdot, \cdot)$ is the Euclidean distance in $\Re^D$. All points in V_i are allocated[2] to the "centre" $f_{\mathcal{N}}(\mathbf{c}_i)$.

We initialize the parameters $\mathbf{W}_{\mathcal{M}}$ of child GTMs $\mathcal{M}$, so that each GTM initially approximates principal component analysis (PCA) of training data in the corresponding Voronoi compartment. For GTM $\mathcal{M}$ corresponding to a compartment V_i, we first evaluate the covariance matrix of training points in V_i and obtain the first L principal eigenvectors. Next, we determine $\mathbf{W}_{\mathcal{M}}$ by minimizing the error

$$E = \frac{1}{2} \sum_{j=1}^{K_{\mathcal{M}}} \| \mathbf{W}_{\mathcal{M}} \ \phi_{\mathcal{M}}(\mathbf{x}_j^{\mathcal{M}}) \ - \ \mathbf{U} \ \mathbf{x}_j^{\mathcal{M}} \|^2, \tag{44}$$

where the columns of $\mathbf{U}$ are the first L principal eigenvectors of the data covariance matrix (see [2]).

Following [2], the parameter $\beta_{\mathcal{M}}$ is initialized to be the larger of the $L + 1$ eigenvalue from PCA, that represents the variance of the data away from the PCA manifold[3], and the square of half of the grid spacing of the PCA-projected latent space centres $\mathbf{x}_j^{\mathcal{M}}$ in the data space.

[2] Ties, as events of measure zero (points that land exactly on the border between the compartments), are broken according to index order.

[3] Alternatively, one can compute the sum of the $D - L$ smallest eigenvalues of the data covariance matrix, divided by $D - L$. This represents the average variance "lost" per discarded dimension and can be shown to be the maximum likelihood estimator for the (isotropic) noise variance in the probabilistic PCA [14].

V. Geometric properties of GTM projection manifolds

We have mentioned in the introduction that allowing for non-linear local projections in the hierarchical visualization system should be accompanied by a set of tools for monitoring the "amount of non-linearity" in the projection manifolds.

Bishop, Svensén and Williams [5] [15] computed local magnification factors of GTM models. The magnification factors describe how small regions of the (low-dimensional) latent space are stretched or compressed when mapped to the (possibly high-dimensional) data space. Similar issues were investigated in the context of SOM e.g. in [16] [17] [18], but such studies are inevitably hampered by the discretized nature of the SOM projection manifold. On the other hand, GTM projection manifold is a smooth function of the latent space coordinates, and so techniques from differential geometry can be used to calculate its geometric properties in a principled way.

Magnification factors represent the extent to which the areas are magnified on projection to the data space. However, when injecting a low dimensional latent space into a high dimensional data space, the projection manifold may form complicated folds that cannot be detected by using magnification factors alone. To provide the user with a tool for monitoring the amount of folding in the projection manifold, we need second-order quantities, such as local curvatures. This in turn, as we shall see in section VII, may be useful for choosing regions of interest when constructing child GTMs, or for updating the regularization parameter of the GTM mapping (see eq. (37)).

In this section, we show how to compute local directional curvatures of the GTM projection manifold and then briefly explain the concept of magnification factors for GTM, as developed in [5] [15].

A. Local directional curvatures

The idea of directional curvature is explained in figure 4. The visualization surface Ω of a GTM $\mathcal{M}$ (see eq. (14)) is the $f_{\mathcal{M}}$–image of the latent space $\mathcal{H}$ and forms an L-dimensional manifold in the data space.

Consider a point $\mathbf{x}_0 \in \mathcal{H}$. Let $\mathbf{x}(b)$, $b \in \Re$, be a straight line passing through $\mathbf{x}_0$ along a unit directional vector $\mathbf{h} = (h_1, h_2, ..., h_L)^T$. The parametric form of $\mathbf{x}(b)$ is given by

$$\mathbf{x}(b) = \mathbf{x}_0 + b\mathbf{h}, \quad b \in \Re. \tag{45}$$

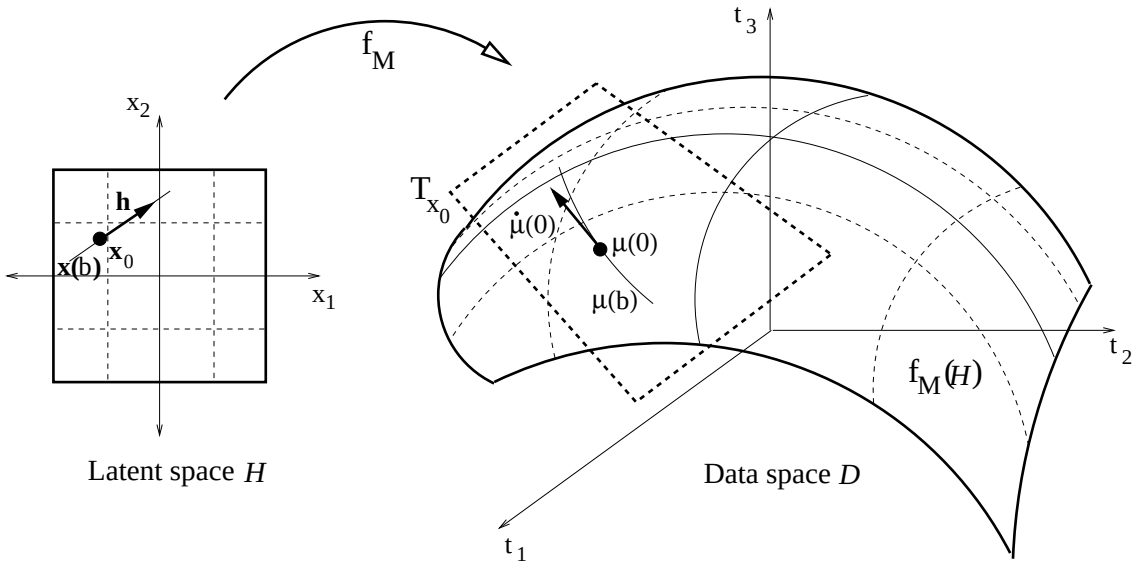

Fig. 4. An explanation of local directional derivative of the projection manifold. A straight line $\mathbf{x}(b)$ passing through the point $\mathbf{x}_0$ in the latent space $\mathcal{H}$ is mapped via $f_{\mathcal{M}}$ to the curve $\mu(b) = f_{\mathcal{M}}(\mathbf{x}(b))$ in the data space. Curvature of μ at $f_{\mathcal{M}}(\mathbf{x}_0) = \mu(0)$ is related to the directional curvature of the projection manifold $f_{\mathcal{M}}(\mathcal{H})$ with respect to the direction $\mathbf{h}$. The tangent vector $\dot{\mu}(0)$ to μ at $\mu(0)$ lies in $\mathbf{T}_{\mathbf{x}_0}$ (dashed rectangle), the tangent plane of the manifold $f_{\mathcal{M}}(\mathcal{H})$ at $\mu(0)$.

As the parameter b varies, the image of the line $\mathbf{x}(b)$ generates the curve

$$\mu(b) = f_{\mathcal{M}}(\mathbf{x}(b)) \tag{46}$$

in the projection manifold Ω, called a lifted line. The tangent to this curve at $f_{\mathcal{M}}(\mathbf{x}_0) = \mu(0)$ is

$$\begin{aligned}
\dot{\mu}(0) &= \left[\frac{\mathrm{d}\ \mu(b)}{\mathrm{d}\ b}\right]_{b=0} \\
&= \left[\sum_{r=1}^{L} \frac{\partial f_{\mathcal{M}}(\mathbf{x})}{\partial x_r}\ \frac{\mathrm{d}\ x_r(b)}{\mathrm{d}\ b}\right]_{\mathbf{x}=\mathbf{x}_0, b=0} \\
&= \sum_{r=1}^{L} \mathbf{\Gamma}_r^{(1)}\ h_r \qquad (47) \\
&= \mathbf{\Gamma}^{(1)}\ \mathbf{h}, \qquad (48)
\end{aligned}$$

where $\mathbf{\Gamma}_r^{(1)}$ is a (column) vector of partial derivatives of the function

$$f_{\mathcal{M}} = (f_{\mathcal{M}}^1, f_{\mathcal{M}}^2, ..., f_{\mathcal{M}}^D)^T, \tag{49}$$

with respect to the r-th latent space variable at $\mathbf{x}_0 \in \mathcal{H}$, and $\mathbf{\Gamma}^{(1)}$ is the $D \times L$ matrix

$$\mathbf{\Gamma}^{(1)} = [\mathbf{\Gamma}_1^{(1)}, \mathbf{\Gamma}_2^{(1)}, ..., \mathbf{\Gamma}_L^{(1)}]. \tag{50}$$

The vectors $\mathbf{\Gamma}_r^{(1)}$, $r = 1, 2, ..., L$, are calculated as follows:

$$\mathbf{\Gamma}_r^{(1)} = \mathbf{W}_{\mathcal{M}} \ \mathbf{\Psi}_r^{(1)}(\mathbf{x}_0) = \mathbf{W}_{\mathcal{M}} \ \left(\frac{\partial \phi_1(\mathbf{x}_0)}{\partial x_r}, \frac{\partial \phi_2(\mathbf{x}_0)}{\partial x_r}, ..., \frac{\partial \phi_{M_{\mathcal{M}}}(\mathbf{x}_0)}{\partial x_r} \right)^T. \tag{51}$$

The tangent vector $\dot{\mu}(0)$ to the lifted line $\mu(b)$ is a linear combination of the columns of $\mathbf{\Gamma}^{(1)}$, and so the range of the matrix $\mathbf{\Gamma}^{(1)}$ is the tangent plane $\mathbf{T}_{\mathbf{X}_0}$ of the projection manifold Ω at $f_{\mathcal{M}}(\mathbf{x}_0) = \mu(0)$. Orthogonal projection onto $\mathbf{T}_{\mathbf{X}_0}$ is a linear operator described by the projection matrix

$$\mathbf{\Pi} = \mathbf{\Gamma}^{(1)} \ \left(\mathbf{\Gamma}^{(1)} \right)^+, \tag{52}$$

where

$$\left(\mathbf{\Gamma}^{(1)} \right)^+ = \left[\left(\mathbf{\Gamma}^{(1)} \right)^T \mathbf{\Gamma}^{(1)} \right]^{-1} \ \left(\mathbf{\Gamma}^{(1)} \right)^T \tag{53}$$

is the Moore-Penrose generalized inverse of $\mathbf{\Gamma}^{(1)}$ (see e.g. [19]).

The second directional derivative [20] of $\mu(b)$ at $\mu(0)$ is

$$\begin{aligned} \ddot{\mu}(0) &= \left[\sum_{s=1}^{L} \frac{\partial}{\partial x_s} \left\{ \sum_{r=1}^{L} \frac{\partial f_{\mathcal{M}}(\mathbf{x})}{\partial x_r} \ h_r \right\} \frac{\mathrm{d} \ x_s(b)}{\mathrm{d} \ b} \right]_{\mathbf{X}=\mathbf{X}_0, b=0} \\ &= \left[\sum_{r=1}^{L} \sum_{s=1}^{L} \frac{\partial^2 f_{\mathcal{M}}(\mathbf{x})}{\partial x_r \partial x_s} \ h_r h_s \right]_{\mathbf{X}=\mathbf{X}_0} \\ &= \sum_{r=1}^{L} \sum_{s=1}^{L} \mathbf{\Gamma}_{r,s}^{(2)} \ h_r \ h_s. \end{aligned} \tag{54}$$

where $\mathbf{\Gamma}_{r,s}^{(2)}$ is a column vector of second-order partial derivatives of $f_{\mathcal{M}}$ with respect to the r-th and s-th latent space variables,

$$\mathbf{\Gamma}_{r,s}^{(2)} = \mathbf{W}_{\mathcal{M}} \ \mathbf{\Psi}_{r,s}^{(2)} = \mathbf{W}_{\mathcal{M}} \ \left(\frac{\partial^2 \phi_1(\mathbf{x}_0)}{\partial x_r \partial x_s}, \frac{\partial^2 \phi_2(\mathbf{x}_0)}{\partial x_r \partial x_s}, ..., \frac{\partial^2 \phi_{M_{\mathcal{M}}}(\mathbf{x}_0)}{\partial x_r \partial x_s} \right)^T. \tag{55}$$

The derivatives are computed at $\mathbf{x}_0 \in \mathcal{H}$.

We decompose the second directional derivative $\ddot{\mu}(0)$ of $f_{\mathcal{M}}$ into two orthogonal components, one lying in the tangent space $\mathbf{T}_{\mathbf{X}_0}$, the other lying in its orthogonal complement $\mathbf{T}_{\mathbf{X}_0}^{\perp}$,

$$\ddot{\mu}(0) = \ddot{\mu}^{\parallel}(0) + \ddot{\mu}^{\perp}(0), \qquad \ddot{\mu}^{\parallel}(0) \in \mathbf{T}_{\mathbf{X}_0}, \ \ddot{\mu}^{\perp}(0) \in \mathbf{T}_{\mathbf{X}_0}^{\perp}. \tag{56}$$

The component $\ddot{\mu}^{\parallel}(0)$ describes changes in the first-order derivatives due to "varying speed of parameterization". Changes in the first-order derivatives that are responsible for curving of the projection manifold Ω are described by the component $\ddot{\mu}^{\perp}(0)$.

By (53) and (54),

$$\begin{aligned} \ddot{\mu}^{\perp}(0) &= (\mathbf{I} - \mathbf{\Pi})\ \ddot{\mu}(0) \\ &= \left[\mathbf{I} - \mathbf{\Gamma}^{(1)} \left(\mathbf{\Gamma}^{(1)}\right)^{+}\right] \left[\sum_{r=1}^{L}\sum_{s=1}^{L} \mathbf{\Gamma}^{(2)}_{r,s}\ h_r\ h_s\right], \end{aligned} \tag{57}$$

where $\mathbf{I}$ is the $D \times D$ identity matrix.

The vector $\ddot{\mu}^{\perp}(0)$ measures the degree to which the visualization manifold Ω (locally) "curves" in the data space manifold $\mathcal{D}$ [21], or speaking in terms of differential geometry (see e.g. [22]), $\ddot{\mu}^{\perp}(0)$ expresses the degree to which Ω is not (locally) autoparallel in $\mathcal{D}$. $\ddot{\mu}^{\perp}(0)$ is the embedding curvature of $\Omega \subset \mathcal{D}$ at $f_{\mathcal{M}}(\mathbf{x}_0)$, evaluated with respect to the latent space direction $\mathbf{h}$.

B. Local magnification factors

For a GTM $\mathcal{M}$, the local magnification factor corresponding to a point $\mathbf{x}_0$ in the latent space $\mathcal{H}$ is the Jacobian $J_{\mathcal{M}}(\mathbf{x})$ of the GTM map $f_{\mathcal{M}}$ (eq. (7)),

$$J_{\mathcal{M}}(\mathbf{x}) = \sqrt{\det(\mathbf{G}_{\mathcal{M}}(\mathbf{x}_0))}, \tag{58}$$

where $\mathbf{G}_{\mathcal{M}}(\mathbf{x}_0)$ is the (local) metric tensor

$$\mathbf{G}_{\mathcal{M}}(\mathbf{x}_0) = \left(\mathbf{\Gamma}^{(1)}\right)^{T} \mathbf{\Gamma}^{(1)}, \tag{59}$$

with $\mathbf{\Gamma}^{(1)}$ defined by (50) and (51). For more details, see [5], [15].

VI. The hierarchical GTM visualization implementation

We organize the plots corresponding to the hierarchy $\mathcal{T}$ of GTMs in a hierarchical tree with the same topology as $\mathcal{T}$. In non-leaf plots, we show the latent space points $\mathbf{c}_i$ that were chosen to be the "centres" of the regions of interest for the child GTMs (see section IV); these are shown as circles labeled by numbers. The numbers determine the order of the corresponding child GTM sub-plots (left-to-right).

We adopt the strategy, suggested in [1], of plotting all the data points on every plot, but modifying the intensity in proportion to the responsibility $P(\mathcal{M}|\ \mathbf{t}_n)$ (see equations (30), (39) and (40)) which each plot (sub-model $\mathcal{M}$) has for the data point $\mathbf{t}_n$. Points that are not well captured by a particular plot will appear with low intensity.

The user can visualize the regions captured by a particular child GTM $\mathcal{M}$, by modifying the plot of its parent, $Parent(\mathcal{M})$, so that instead of the parent responsibilities, $P(Parent(\mathcal{M})|\ \mathbf{t}_n)$, the responsibilities of the model $\mathcal{M}$, $P(\mathcal{M}|\ \mathbf{t}_n)$, are used. In our software, this is done by simply clicking with a mouse on a chosen child GTM plot. Alternatively, the user can modulate with responsibilities $P(Parent(\mathcal{M})|\ \mathbf{t}_n)$ all the ancestor plots up to *Root*, i.e. all plots appearing in $Path(Parent(\mathcal{M}))$ (see section II)[4]. The chosen child plot is highlighted by a bold red frame. The ancestor plots appear in bold green frames. The rest of the plots show data projections as low-intensity gray points. As will be shown in section VII, such a modulation of ancestor plots is an important tool to help the user relate child plots to their parents.

The hierarchical structure used for plotting the GTMs' projections is also used to show the magnification factors of GTMs in the hierarchy. For every GTM $\mathcal{M}$, we evaluate the local magnification factor $J_{\mathcal{M}}(\mathbf{x})$ (eq. (58)) in each latent space centre $\mathbf{x}_i^{\mathcal{M}}$, $i = 1, 2, ..., K_{\mathcal{M}}$ (see section III). The intensities with which the magnification factors are shown are scaled with respect to the minimal and maximal magnification factors in the whole hierarchy. The scale is shown as a color bar near the top visualization plot corresponding to the root GTM. The user can get a locally scaled plot of magnification factors by clicking on a chosen plot corresponding to a local GTM $\mathcal{M}$. Magnification factors of the GTM $\mathcal{M}$ are then shown scaled with respect to the minimal and maximal magnification factors of $\mathcal{M}$. A local scaled color bar is also provided.

Finally, the philosophy for showing the local directional curvatures is the same as that for showing the magnification factors. First, the number N_h of different latent space directions $\mathbf{h}$, with respect to which the curvatures will be computed is determined (see section V-A). In the case of two-dimensional latent space, the directions $\mathbf{h}_j$, $j = 1, 2, ..., N_h$, correspond to the N_h equidistant points on the unit circle, subject to the constraint that the first direction is $(1, 0)$. For every GTM $\mathcal{M}$, we evaluate the (Euclidean) norm of the directional curvature $\ddot{\mu}^{\perp}(0)$ (eq. (57)) at each latent space centre $\mathbf{x}_i^{\mathcal{M}}$, with respect to all directions $\mathbf{h}_j$, $j = 1, 2, ..., N_h$. In the final plot, we show, for each latent space centre $\mathbf{x}_i^{\mathcal{M}}$, the maximal norm of the curvature across the different "probing" directions $\mathbf{h}_j$, $j = 1, 2, ..., N_h$. The direction of the maximal curvature corresponding to a latent space

[4]Thanks to one of the reviewers for this suggestion.

centre $\mathbf{x}_i^{\mathcal{M}}$ is shown as a black line of length proportional to the curvature's norm. As in the case of magnification factors, the intensity of curvatures in the hierarchy of GTMs is scaled by the minimal and maximal curvatures found in the whole hierarchy. A locally scaled plot of curvatures can be obtained by clicking on a chosen plot corresponding to a local GTM.

The software has been written in MATLAB and is available from
`http://www.ncrg.aston.ac.uk/netlab/`.

VII. EXPERIMENTS

In this section we illustrate the hierarchical GTM visualization algorithm on a toy data set and two more complex data collections.

Although the algorithm is derived in a general setting in which individual GTMs $\mathcal{M}$ in the hierarchy can have different sets of latent space centres $\mathbf{x}_i^{\mathcal{M}}$, $i = 1, 2, ..., K_{\mathcal{M}}$, and basis functions ϕ_j, $j = 1, 2, ..., M_{\mathcal{M}}$, in the experiments reported here, we used a common GTM configuration for all models in the hierarchy. In particular, the latent space $\mathcal{H}$ was taken to be the two-dimensional interval $\mathcal{H} = [-1, 1] \times [-1, 1]$, the latent space centres $\mathbf{x}_i^{\mathcal{M}} \in \mathcal{H}$ were positioned on a regular 15×15 square grid and there were 16 basis functions ϕ_j centered on a regular 4×4 square grid. The basis functions were spherical Gaussian functions of the same width $\sigma = 1.0$. We account for a bias term by using an additional constant basis function $\phi_{17}(\mathbf{x}) = 1$, for all $\mathbf{x} \in \mathcal{H}$. The regularization coefficient $\alpha_{\mathcal{M}}$ was set to 0.1.

For each model $\mathcal{M}$ in the hierarchy, the directional curvatures (57) were evaluated in all latent space centres $\mathbf{x}_i^{\mathcal{M}}$ along $N_h = 16$ "probing" directions $\mathbf{h}_j$ (see section VI).

A. *Toy data*

The first experiment was conducted with a toy data set of 3000 points $\mathbf{t} = (t_1, t_2, t_3)^T$ lying on a two-dimensional manifold in the three-dimensional space. The manifold is shown in figure 1 and is described by

$$t_3 = 2 \sum_{c_1, c_2 \in \{-2, 2\}} \exp\left\{-(t_1 - c_1)^2 - (t_2 - c_2)^2\right\}, \quad (t_1, t_2) \in [-4, 4]^2. \tag{60}$$

To demonstrate the hierarchical GTM algorithm, we associated the points in the four "humps" with four different classes, $\mathcal{C}_i$, $i = 1, 2, 3, 4$. After training a top level GTM, we

constructed a mixture of GTMs on four regions of interest centered at the four humps. Each GTM in the mixture was supposed to fit the distribution of the corresponding hump class. Figure 5 shows projection manifolds corresponding to the mixture of four GTMs.

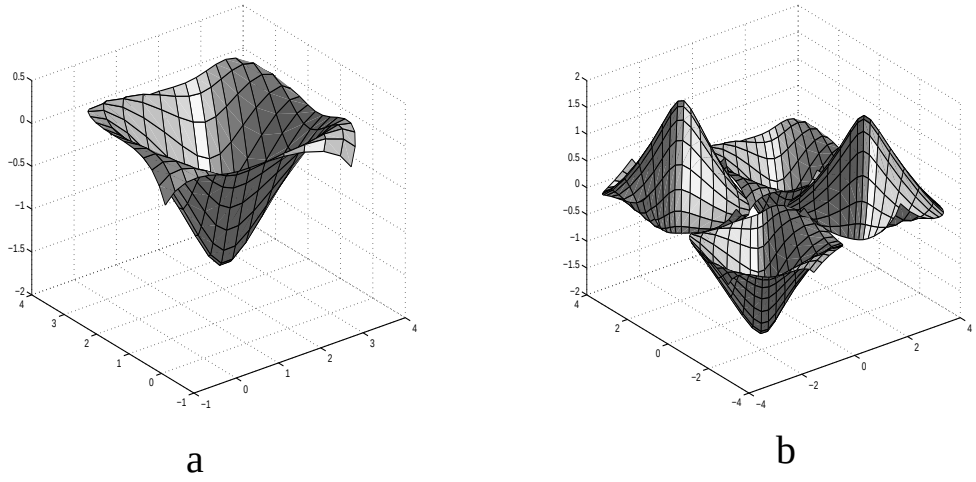

Fig. 5. Projection manifolds in data space of the second-level GTMs trained on the toy data. Shown is a collection of all second-level projection manifolds (b), as well as the projection manifold of a single mixture component modeling the "hump" centred at $(t_1, t_2) = (2, 2)$ (a).

Data projections realized by the hierarchy are presented in figure 6. By clicking on the third second-level model $\mathcal{M}$, point intensities in the visualization plot of its parent, $Parent(\mathcal{M}) = Root$, are modulated by the second-level model responsibilities $P(\mathcal{M}|\ \mathbf{t}_n)$ (see section VI).

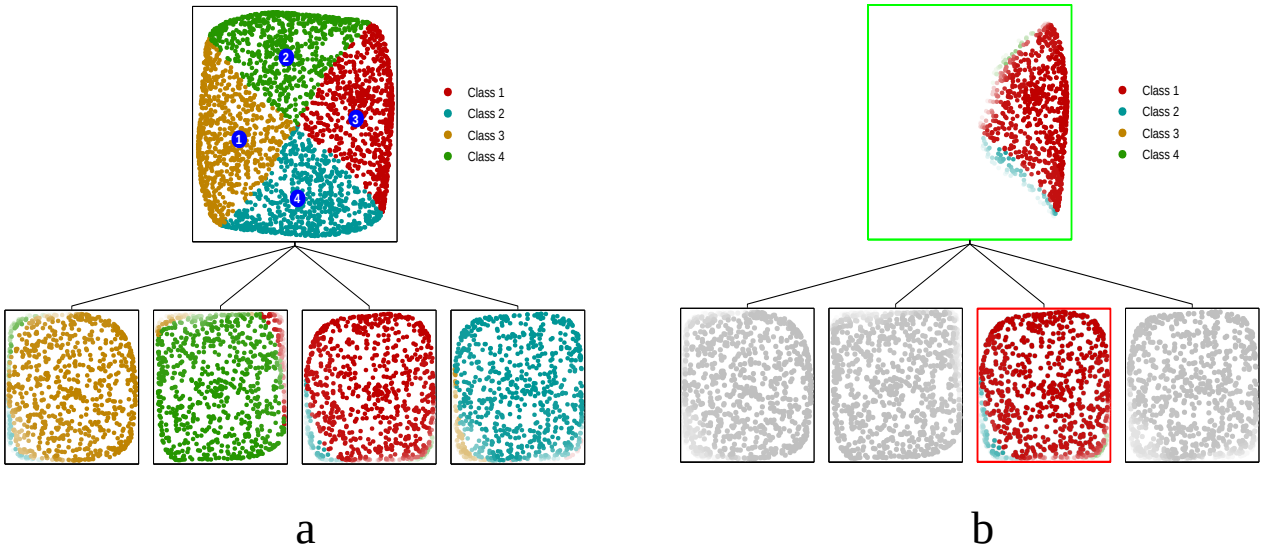

Fig. 6. (a) – The complete visualization plot for the toy data. (b) – Points captured by the third model at the second level of the hierarchy are shown in the top-level plot.

Magnification factors and curvatures of the hierarchy of GTMs are shown in figures 7(a) and 7(b), respectively. In this case, the magnification factors and curvatures are almost complementary. When mapped into the projection manifold, the latent space is mostly stretched in the *Root* model, while the dominant curvatures were detected at the second level of the hierarchy. Note how the curvature near the edges and at the "peak" of the second-level models (see figure 5) is reflected in the curvature plot (figure 7 (b)).

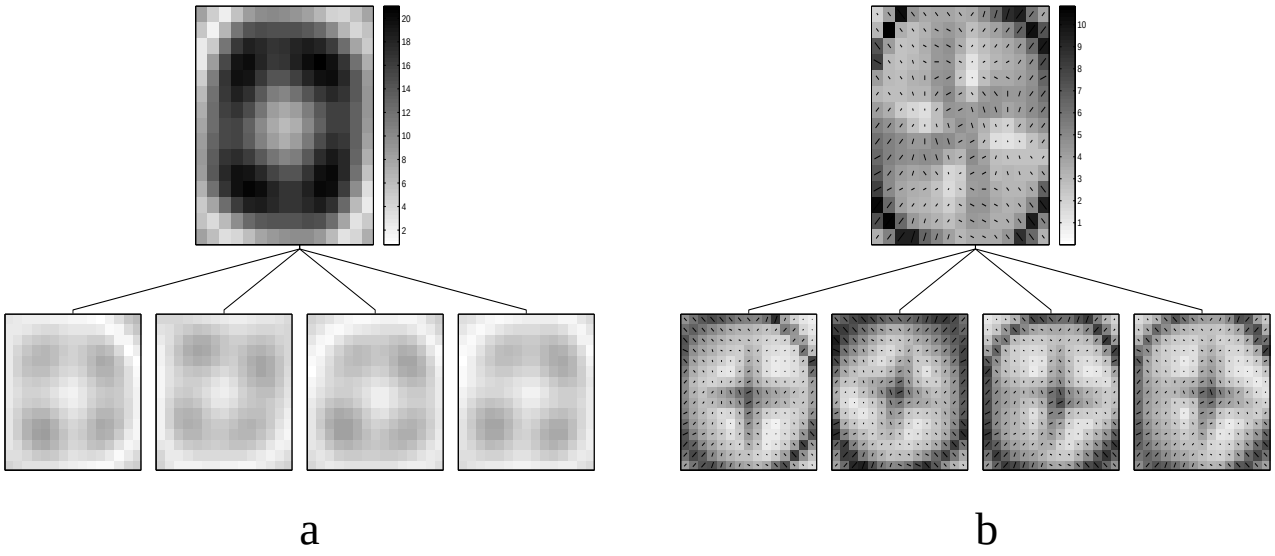

Fig. 7. Magnification factors (a) and curvatures (b) computed on projection manifolds of the hierarchical GTM trained on the toy data.

B. Oil flow data

The oil flow data set[5] was used to demonstrate the locally linear hierarchical visualization algorithm of Bishop and Tipping [1], called PhiVis[6]. This 12-dimensional data set arises from a physics-based simulation of non-invasive monitoring system, used to determine the quantity of oil in a multi-phase pipeline containing a mixture of oil, water and gas. The data set consists of 1000 points obtained synthetically by simulating the physical process in the pipe. Points in the data set are classified into three different multi-phase flow configurations, namely *homogeneous*, *annular* and *laminar*. Data is distributed in numerous distinct clusters and is expected to have (locally) an intrinsic dimensionality of two [1].

A hierarchy of GTMs up to level 4 was trained on this data set and the final visualization

[5]The oil flow data set can be obtained from `http://www.ncrg.aston.ac.uk/GTM/3PhaseData.html`.

[6]A MATLAB code for PhiVis is publicly available at `http://www.ncrg.aston.ac.uk/PhiVis/`.

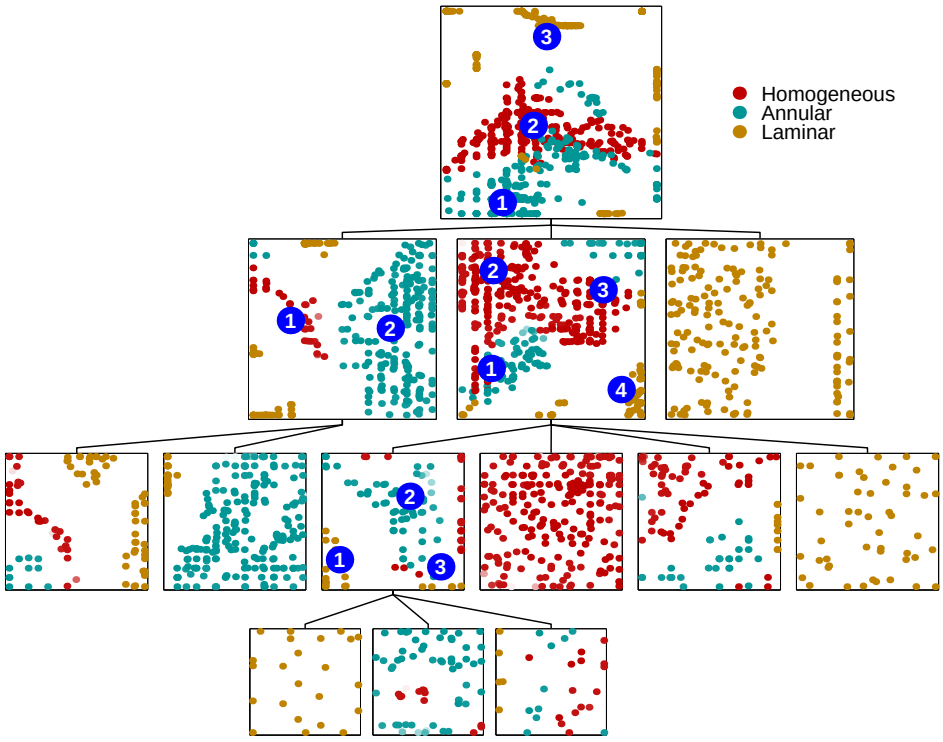

Fig. 8. A complete visualization plot for the oil flow data given by the hierarchy of GTMs. The projections are colored according to the class of the corresponding data points: homogeneous – red, annular – blue, laminar – yellow.

plot can be seen in figure 8. The corresponding magnification factor and directional curvature plots are shown in figures 10 and 12, respectively. The curvature plot of the root GTM reveals that the two-dimensional projection manifold folded three times in order to "capture" the distribution of points in the 12-dimensional space. Interestingly, the three multi-phase flow configurations seem to be roughly separated by the folds (compare the top level visualization plot in figure 8 with the corresponding curvature plot in figure 12). We confirmed this hypothesis by constructing three local second-level visualization plots initiated in the regions between the folds. Curvature and magnification factor plots of the lower level GTMs reveal that, compared with the root GTM, the lower level projection manifolds do not significantly stretch/contract and are almost flat. Figure 11 was obtained by clicking on the first plot at level three of the hierarchy and shows a detailed portrait of local magnification factors of the selected model.

By clicking on the first level-four GTM $\mathcal{M}$ modeling laminar flow points, we can trace the position of points locally captured by $\mathcal{M}$ in the visualization plots of all its ancestors

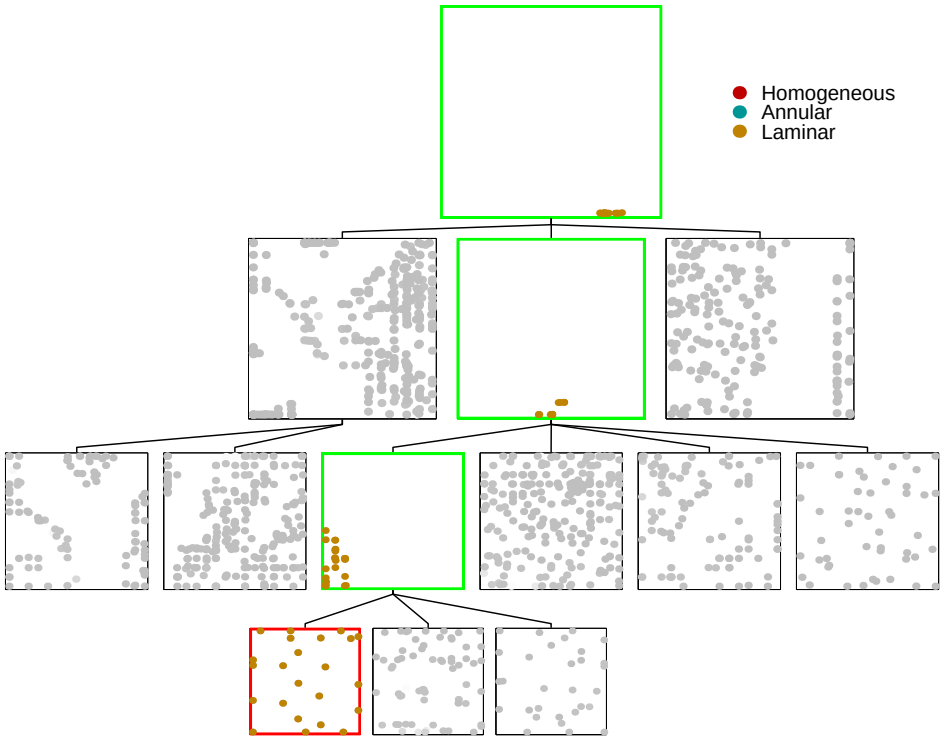

Fig. 9. A hierarchical visualization plot for the oil flow data, in which the set of points captured by the first GTM at level 4 of the hierarchy (red border) is highlighted in the visualization plots of all its ancestors (green borders).

(see figure 9). Clicking on sub-plots in the visualization hierarchy $\mathcal{T}$ and comparing the child-modulated ancestor plots (see section VI) with the full visualizations in figure 8 is a valuable tool for understanding the relationship among the individual plots in the hierarchy $\mathcal{T}$.

For comparison, we show in figure 13 a complete four-level hierarchical visualization plot for the locally linear system PhiVis. When the plot corresponds to a leaf model, PhiVis copies the plot to lower levels. In addition to data projections, visualization plots of models that have children show the orthogonal projections of the child visualization planes onto the parent visualization plane.

In the hierarchical GTM visualization, we get an almost perfect separation of points into the three classes even in the top level plot. Indeed, looking at figures 10 and 12 we see that most stretching and folding is detected in the *Root* GTM. The lower level GTMs inject their latent space into the data space without much deformation, suggesting that the local distribution of points is roughly two-dimensional and flat. This confirms the intuition that led Bishop and Tipping to use the oil flow data to demonstrate the PhiVis

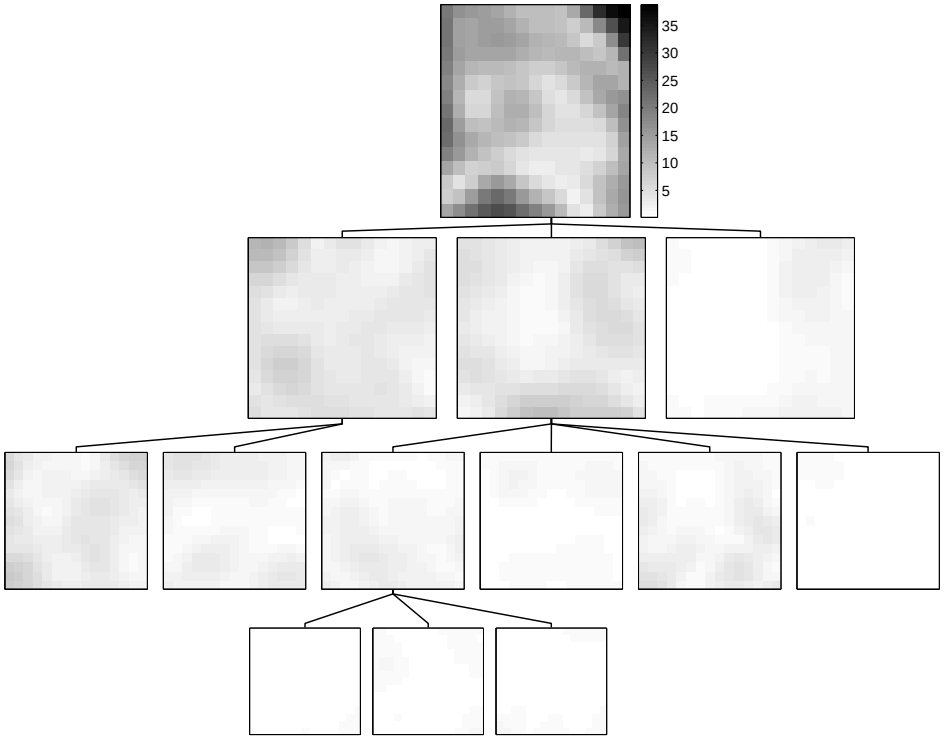

Fig. 10. A visualization plot of magnification factors in the hierarchy of GTMs fitted on the oil flow data.

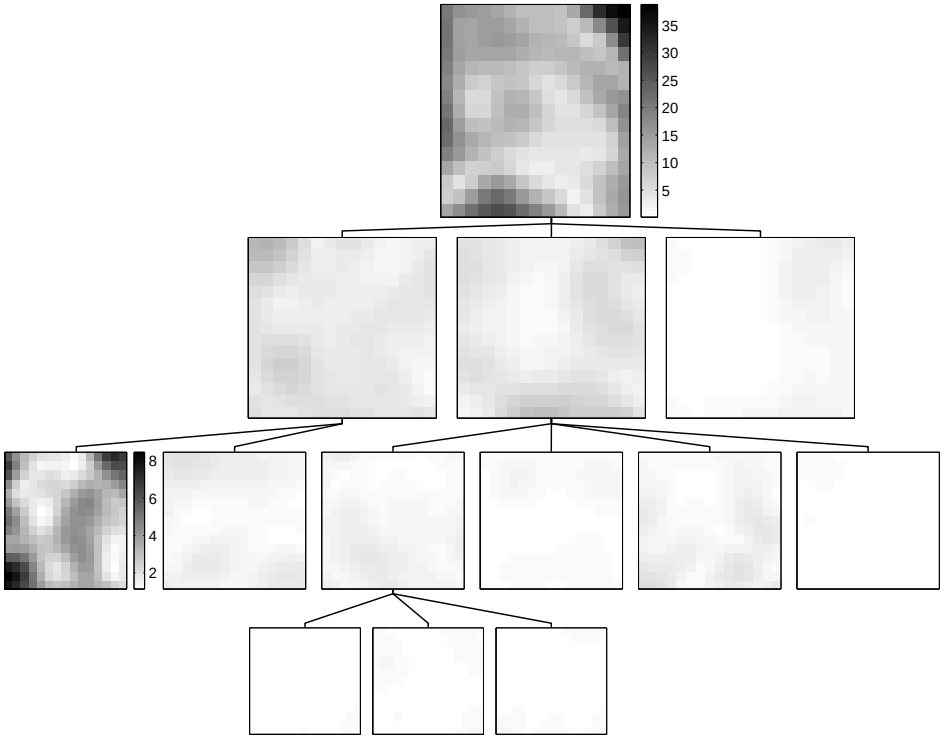

Fig. 11. A hierarchical visualization of magnification factors showing a detailed portrait of local magnification factors of the first GTM at level 3.

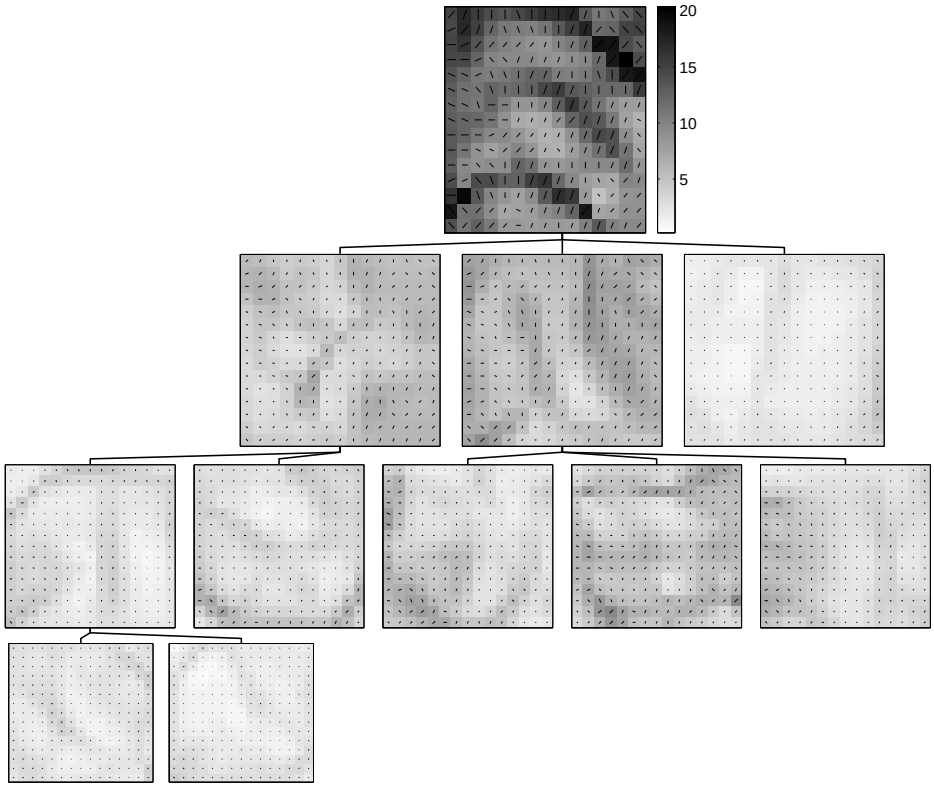

Fig. 12. A hierarchical visualization of local curvatures in the hierarchy of GTMs fitted on the oil flow data.

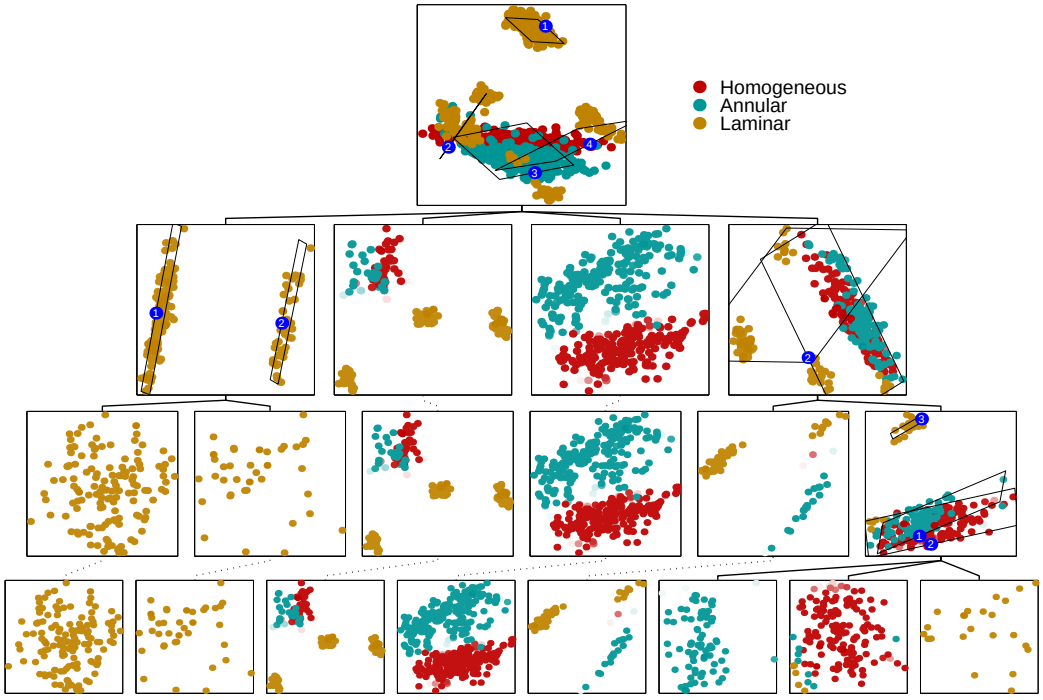

Fig. 13. A hierarchical visualization of the oil flow data given by the locally linear system PhiVis.

visualization system [1].

C. Image segmentation data

In the last experiment we visualize image segmentation data[7] obtained by randomly sampling patches of 3x3 pixels from a database of 7 outdoor images. The patches are characterized by 18 continuous attributes and are classified into 7 classes: *brickface, sky, foliage, cement, window, path* and *grass*. The data set contains 2310 18-dimensional points, 330 instances per class. We merged the original seven classes into four composite classes: *cement + path*, *brickface + window*, *grass + foliage* and *sky*.

We trained a four-level hierarchy of GTMs on the image segmentation data and the resulting projection, magnification factor and curvature plots are presented in figures 14, 16 and 17, respectively.

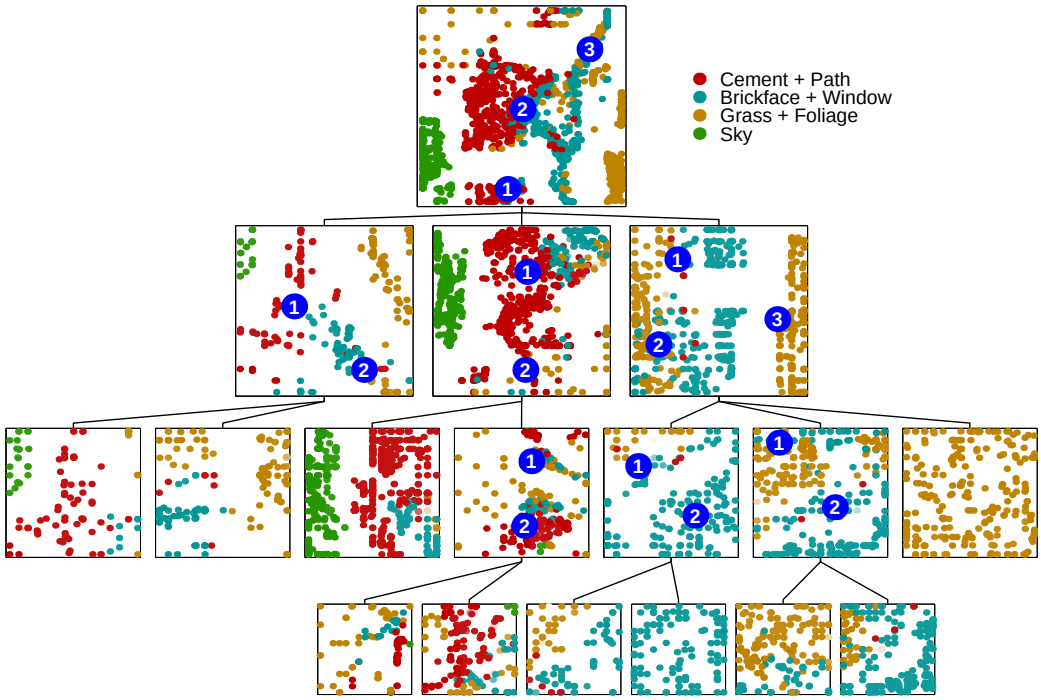

Fig. 14. A hierarchical GTM visualization plot of image segmentation data.

A hierarchical visualization of the image segmentation data given by PhiVis is shown in figure 18.

In contrast to the the oil flow experiment, the image segmentation data is difficult to capture using PhiVis. As seen in figures 16 and 17, very strong local stretchings and highly

[7] The image segmentation data set can be downloaded from the Delve repository `http://www.cs.utoronto.ca/~delve/data/datasets.html`.

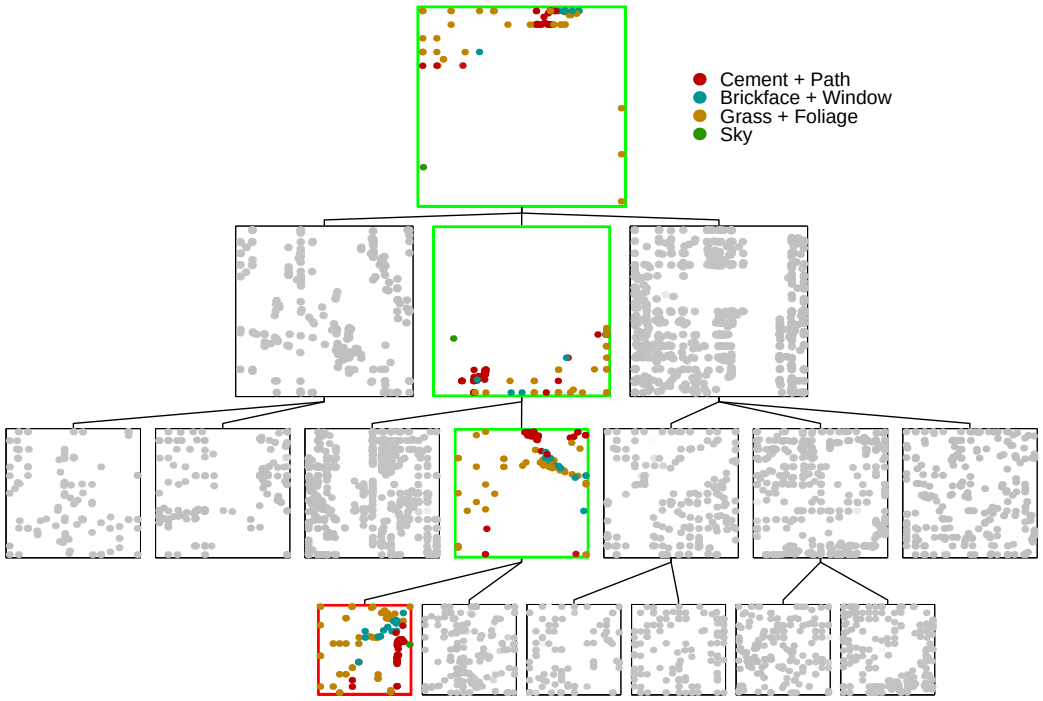

Fig. 15. A hierarchical GTM visualization plot of image segmentation data in which the set of points captured by the first level-four GTM (red border) is highlighted in the visualization plots of its ancestor GTMs (green borders).

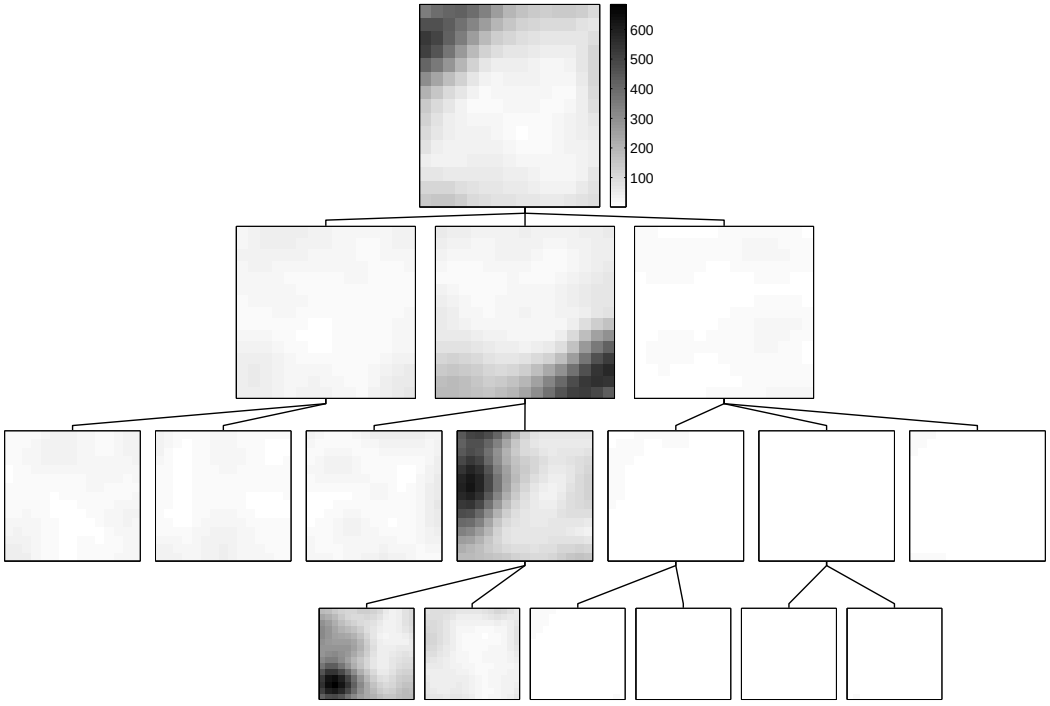

Fig. 16. Magnification factors in a hierarchy of GTMs fitted on image segmentation data.

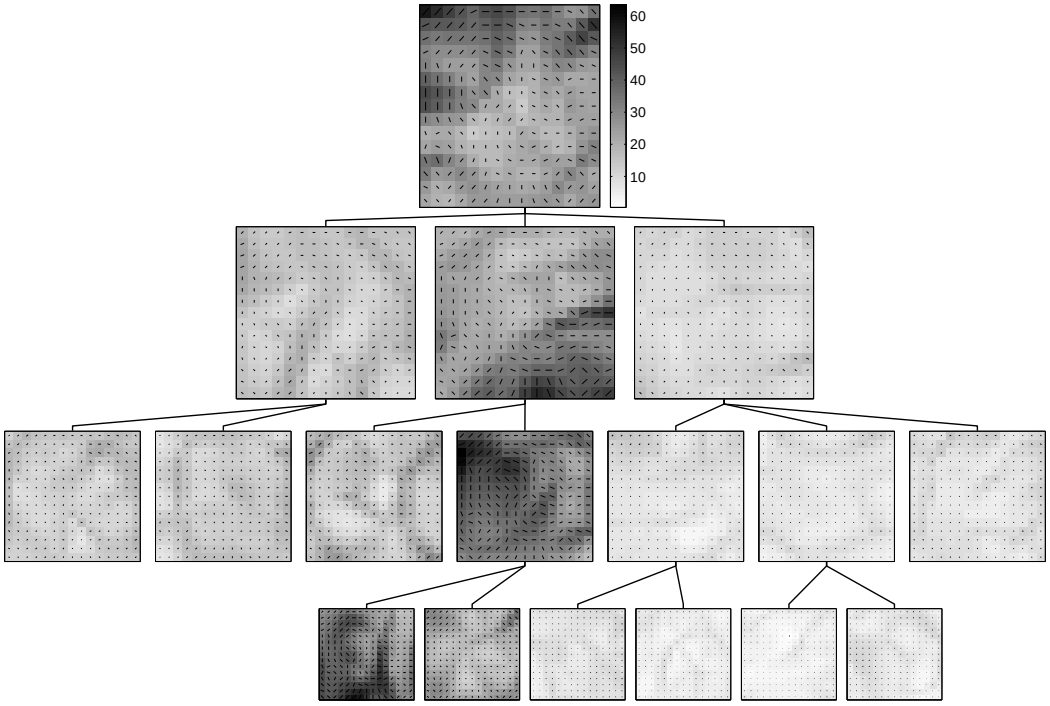

Fig. 17. Local curvatures of projection manifolds in a hierarchy of GTMs trained on image segmentation data.

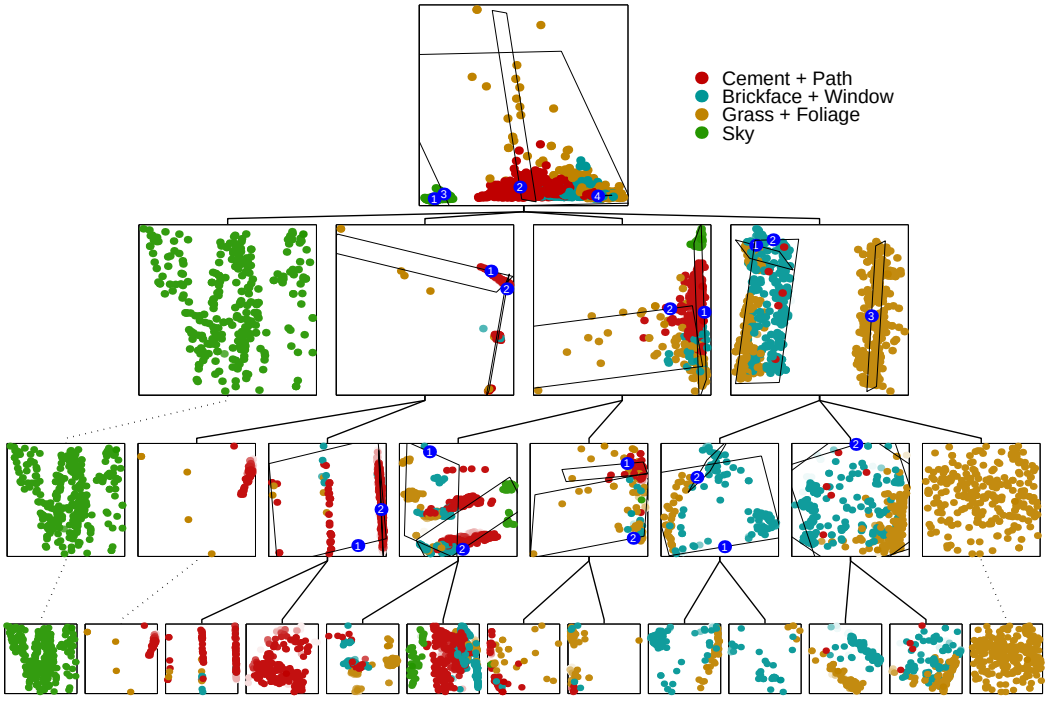

Fig. 18. A hierarchical visualization of the image segmentation data given by PhiVis.

curved visualization surfaces throughout the hierarchy of GTMs are needed to capture the data characterized by higher intrinsic dimensionality and the presence of "outliers". Note the highly non-linear nature of the sequence of GTMs on the path from the first level-four model to the *Root*. The *Root* GTM had to stretch a long way in order to capture the grass+foliage points appearing near the top left corner of the *Root* visualization plot in figure 14. Actually, these points caused most of the linear data projections in the top level PhiVis plot to cluster near the bottom of the plot.

Looking at figures 16 and 17 we can see dominant stretchings and foldings in the second level-two GTM, fourth level-three GTM and the first GTM on level four. The areas of high magnification and curvature in these plots correspond to the areas containing projections of the "outlier" grass+foliage points detected in the top level plot. This is confirmed by the child-modulated ancestor plot technique, illustrated in figure 15, highlighting the position of points captured by the first level-four GTM in its ancestor plots.

VIII. Discussion

We have extended the locally linear hierarchical visualization system PhiVis proposed in [1] to allow for non-linear projection manifolds. Our system is statistically principled and is built using the EM algorithm in a top-down fashion. The authors of PhiVis emphasize that there is no objective measure of quality in data visualization, but argue that a hierarchical visualization model can be a very useful tool for the visualization and exploratory analysis of data in many applications [1].

Our experiments suggest that by allowing for non-linearity in the projection manifolds, we can indeed create more detailed and parsimonious visualization plots. While Principal Component Analysis (PCA) can introduce in the visualization plot only "global" stretchings along the principal axes, the non-linear projection manifold of GTMs can locally stretch and fold in the data space. This enables our system to make full use of the latent space when describing the local distributions of points. As a result, the PhiVis plots are often characterized by dense isolated clusters. This phenomenon is not seen in our system.

Of course, we can always "reasonably" cover the low-dimensional non-linear data manifold by using enough local linear patches, but, as we saw in the last section, this can often lead a visualization hierarchy that is much more complex and difficult to read. This

in turn makes it difficult for the user to grasp and understand the overall layout of data points in a high-dimensional space. Non-linearity in the projection manifolds allows us to construct more parsimonious visualization hierarchy, but there is a price to pay. It is no longer possible to relate children plots to the corresponding parent plots simply by showing projections of the image of the children latent space in the parents' latent space, as we have seen in the linear system PhiVis. One could consider projecting the image of the latent grid of the children GTMs onto the latent space of the parent, but, multi-modalities in the posterior distribution over the parent latent space would make interpretation of such plots problematic[8].

There are several tools implemented in our hierarchical non-linear visualization system that can help the user to understand the visualization plots and, if needed, further refine the visualization hierarchy:

1. The user can highlight in the ancestor plots the data points which are under responsibility of a selected child plot. This illustrates the history of projections in higher level plots of points captured by a lower level plot.

2. Although not reported here, we have extended our system to identify points, e.g. by their index in the data set, by clicking on their projections in a chosen plot. This way the user can relate lower level plots with their ancestors in a more detailed manner.

3. The smooth character of the GTM mapping from the latent space to the data space makes it possible to calculate local stretching and folding characteristics of the non-linear projection manifolds. The low dimensional projection manifold can form complicated folds and/or significant contractions/stretchings in the high-dimensional data space. Considering the projection plot alone, it is difficult to judge the actual "layout" of points in the data space. For example, regions of high contraction of the visualization manifold often correspond to regions of dense clusters in the data space, whereas highly stretched areas usually fill the space between the clusters [15]. Without this additional information, the users may not realize that the almost homogeneous group of points they see on the visualization plot actually comes from several well-separated clusters. Also, local curvature patterns in the projection manifold provide information about dominant folds. This, together with the contraction/expansion characterization of the manifold, can be helpful

[8]We are thankful to one of the reviewers for bringing up this point.

in determining the "regions of interest" for constructing local sub-plots in the hierarchy of visualization plots.

It should be mentioned that GTM requires the specification of the hyperparameters σ (width of the Gaussian basis functions) and α (regularization coefficient for weights $\mathbf{W}$). Both hyperparameters determine the "stiffness" of the projection manifold. In this study we follow recommendation of [2] to set σ to $\sigma = 2s$, where s is the distance between two neighboring centres of the basis functions. Bayesian inference of the GTM hyperparameters, introduced in [4], would enormously prolong training of local models in our visualization hierarchy. However, since we do not rely on a single "top-level" visualization plot, as long as the projection manifolds are "reasonably" smooth, and we can monitor the amount of stretching and folding by inspecting the local magnification factor and directional curvature plots, one expects to obtain good representations of the local data distributions at lower levels of the visualization hierarchy.

Our hierarchical GTM visualization system works in an interactive way: based on lower level projections, regions of interest for higher level models are determined by the user. Algorithms for self-consistent fitting of the hierarchical tree can be easily created by employing some form of hierarchical clustering, e.g. hierarchical clustering of data by deterministic annealing [23]. However, the user-driven construction of the hierarchical visualization plot is a natural candidate for investigation of the data via low-dimensional projections.

IX. Conclusion

The main contributions of the paper can be summarized as follows:

1. We have extended the locally linear hierarchical visualization system PhiVis proposed by Bishop and Tipping [1] to allow for non-linear projection manifolds. Like PhiVis, our system is statistically principled and is built interactively in a top-down fashion using the EM algorithm.

2. We further extended the work presented in [1] by introducing a general formulation of a hierarchical probabilistic model consisting of local probabilistic models organized in a hierarchical tree. General training equations are derived, regardless of the position of the model in the tree.

3. We have exploited the smooth character of GTM projection manifold to derive expressions for the local directional curvatures of the manifold.

4. We have built an interactive system for non-linear hierarchical data visualization that enables the user to

(a) better understand the visualization hierarchy by highlight the data in the ancestor visualization plots that are captured by a child GTM;

(b) visualize the magnification factor structure across the hierarchy of GTMs, as well as to interactively check a detailed magnification factor layout of a chosen local model;

(c) visualize in a similar manner the structure of local directional curvatures of the projection manifolds.

Such information can useful for further refinement of the hierarchical visualization plot, as well as for controlling the amount of regularization imposed on local models.

Acknowledgments

The authors would like to thank Yi Sun for helpful discussions and the anonymous reviewers for many useful suggestions that helped to improve presentation of the paper. MATLAB implementation of the hierarchical GTM visualization system is partly based on the MATLAB implementation of PhiVis by Bishop and Tipping.

References

[1] C.M. Bishop and M.E. Tipping, "A hierarchical latent variable model for data visualization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 281–293, 1998.

[2] C.M. Bishop, M. Svensén, and C.K.I. Williams, "GTM: The generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–235, 1998.

[3] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1479, 1990.

[4] C.M. Bishop, M. Svensén, and C.K.I. Williams, "Developments of the Generative Topographic Mapping," *Neurocomputing*, vol. 21, pp. 203–224, 1998.

[5] C.M. Bishop, M. Svensén, and C.K.I. Williams, "Magnification factors for the SOM and GTM algorithms," in *Proceedings 1997 Workshop on Self-Organizing Maps, Helsinki, Finland*, 1997.

[6] R. Miikkulainen, "Script recognition with hierarchical feature maps," *Connection Science*, vol. 2, pp. 83–101, 1990.

[7] C. Versino and L.M. Gambardella, "Learning fine motion by using the hierarchical extended Kohonen map," in *Proceedings ICANN96, International Conference on Artificial Neural Networks, Bochum, Germany*, 1996, pp. 221–226.

[8] C. Versino and L.M. Gambardella, "Learning fine motion in robotics: Experiments with the hierarchical extended Kohonen map," in *Proceedings ICONIP96, International Conference on Neural Information Processing, Hong Kong*, 1996, vol. 2, pp. 921–925.

[9] C.K.I. Williams, "A MCMC approach to hierarchical mixture modelling," in *Advances in Neural Information Processing Systems 12*, S. Solla, T. Leen, and K.R. Muller, Eds., pp. 680–686. MIT Press, 2000.

[10] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[11] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, B*, vol. 39, no. 1, pp. 1–38, 1977.

[12] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, England, 1988.

[13] F. Aurenhammer, "Voronoi diagrams – survey of a fundamental geometric data structure," *ACM Computing Surveys*, , no. 3, pp. 345–405, 1991.

[14] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, pp. 443–482, 1999.

[15] C.M. Bishop, M. Svensén, and C.K.I. Williams, "Magnification factors for the GTM algorithm," in *Proceedings IEE Fifth International Conference on Artificial Neural Networks*. 1997, pp. 64–69, IEE, London.

[16] A. Ultsch and H.P. Siemon, "Kohonen's self organizing feature maps for exploratory data analysis," in *Proceedings of International Neural Network Conference (INNC'90)*, Dordrecht, Netherlands, 1990, pp. 305–308, Kluwer.

[17] A. Ultsch, "Knowledge extraction from self-organizing neural networks," in *Information and Classification*, O. Opitz, B. Lausen, and R. Klar, Eds., pp. 301–306. Berlin: Springer, 1993.

[18] J. Vesanto, "SOM-based data visualization methods," *Intelligent Data Analysis*, vol. 3, pp. 111–126, 1999.

[19] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, England, 1985.

[20] G.A.F. Seber and C.J. Wild, Eds., *Nonlinear Regression*, John Wiley and Sons, New York, NY, 1989.

[21] D.M. Bates and D.G. Watts, "Relative curvature measures of nonlinearity (with discussion)," *J. R. Stat. Soc. B*, vol. 42, pp. 1–25, 1980.

[22] S.-I. Amari, *Differential-Geometrical Methods in Statistics*, Springer-Verlag, Berlin, 1985.

[23] K. Rose, E. Gurewitz, and G.C. Fox, "Statistical mechanics and phase transitions in clustering," *Physical Review Letters*, vol. 65, no. 8, pp. 945–948, 1990.