Optimal smoothing of kernel-based topographic maps with application to density-based clustering of shapes

Marc M. Van Hulle & Temujin Gautama
K.U.Leuven, Laboratorium voor Neuro- en Psychofysiologie
Campus Gasthuisberg, Herestraat 49, B-3000 Leuven, BELGIUM
E-mail: {marc,temu}@neuro.kuleuven.ac.be

June 17, 2002

Abstract

A crucial issue when applying topographic maps for clustering purposes is how to select the map's overall degree of smoothness. In this paper, we develop a new strategy for optimally smoothing, by a common scale factor, the density estimates generated by Gaussian kernel-based topographic maps. We also introduce a new representation structure for images of shapes, and a new metric for clustering them. These elements are incorporated into a hierarchical, density-based clustering procedure. As an application, we consider the clustering of shapes of marine animals taken from the SQUID image database. The results are compared to those obtained with the CSS retrieval system developed by Mokhtarian and co-workers, and with the more familiar Euclidean distance-based clustering metric.

Keywords: shape clustering, kernel-based topographic maps, density estimation, density-based clustering, optimal smoothing

1 Introduction

The visualization of clusters in high-dimensional spaces with topographic maps has recently attracted the attention of the data mining community (Deboeck, 1998; Cottrell *et al.*, 1999; Lagus and Kaski, 1999, Vesanto, 1999; Vesanto and Alhoniemi, 2000; Himberg *et al.*, 2001, and references therein). Topographic maps can be regarded as discrete lattice-based approximations to non-linear data manifolds, and, in this way, used for projecting and visualizing the data. What is less evident from the literature, is that this required a shift in the clustering

paradigm. Originally, topographic maps, such as the popular Self-Organizing Map (SOM) (Kohonen, 1982, 1984, 1995), were used for *similarity-based* clustering: the converged neuron weights correspond to the centers of the individual clusters, and the Voronoi regions defined by them correspond to the cluster regions. Data points are assigned to the neurons for which the Euclidean distances to their weights are minimal (Winner-Take-All neurons)[1]. However, this requires prior knowledge of the number of clusters in the data set, which makes this approach less suited for an exploratory data analysis. Furthermore, similarity-based clustering assumes, albeit often tacitly, that the cluster shape is hyperspherical, at least when the Euclidean distance metric is used. The shape of a real-world cluster may not comply with this assumption.

There is another way by which clustering can be performed with topographic maps, and which is deemed more suitable for exploratory data analysis. The distribution of the converged neuron weights can be regarded as a non-parametric estimate of the data density. The high density regions are then hypothesized to correspond to individual clusters. This approach is called *density-based* clustering. Unfortunately, the weight density achieved with the SOM algorithm at convergence is not a linear function of the data density (Ritter, 1991, Dersch and Tavan, 1995), so that the quality of the density estimate is often inferior to what can be expected from other techniques, including several topographic map formation algorithms (for a review, see Van Hulle, 2000a). One way to improve the density estimation capabilities is to employ neurons with kernel-based activation functions, such as Gaussians, instead of Winner-Take-All functions. Several versions of such kernel-based topographic maps, as they are called, have been suggested in the literature, together with a series of learning algorithms for training them (for an overview, see Van Hulle, 2002a). In an attempt not to assume prior knowledge of the number of clusters, and since the weight distribution achieved at convergence is related to the input density, several authors have developed ways of visualizing clusters in the input density using topographic maps (see Kohonen, 1995, p. 117). A simple technique, called *gray level clustering*, represents the relative distances between the weights of neighboring neurons, by gray scales.

There are two key issues we have to deal with when using topographic maps for density-based clustering purposes: topological defects and optimal smoothing of the density estimate. Indeed, when the topographic map contains topological defects – neighboring data points are not mapped onto neighboring neurons – a contiguous cluster could become split into separate clusters. In previous contributions (Van Hulle, 2000b; Van Hulle and Gautama, 2002b),

[1] In fact, when used in batch mode, the SOM algorithm has an intimate connection with the classic k-means clustering algorithm (MacQueen, 1967; Krishnaiah and Kanal, 1982) (see also Kohonen, 1995, p. 127).

also to this journal, we have introduced an algorithm that monitors the degree of topology preservation achieved by kernel-based maps during learning. As a real-world application, we considered the identification of musical instruments, and the notes played by them, by means of a hierarchical clustering analysis, starting from the music signal's spectrogram. Topographic map formation was achieved with the kernel-based Maximum Entropy learning Rule (kMER), and it was shown to yield an equiprobabilistic map of heteroscedastic Gaussian density mixtures (Van Hulle, 1998, 2000a).

In this paper, we will concentrate on the second issue and develop a new technique for determining the overall degree of smoothness of the density estimate by scaling the widths of all kernels by a common factor. As a real-world application, we will consider the image database of contours of marine animals, compiled by Mokhtarian and co-workers (Mokhtarian *et al.*, 1996), and perform a hierarchical clustering in order to group contours that show similar global shapes. We will introduce for this purpose a new clustering metric for assigning contours to clusters ("labeling"). The metric is based on outlier detection.

The paper is organized as follows. First, we briefly re-introduce the kMER learning scheme that we will use for training the kernel-based topographic maps. Then, in section 3, we show how density estimation can be performed with these maps, introduce the issue of optimal smoothness, and our solution for determining it. In section 4, we describe the image database of marine animal contours, introduce our contour representation, and formulate the clustering problem. In section 5, we detail our hierarchical clustering procedure as it is applied to the image database. We also introduce here our new labeling method based on outlier detection. Finally, we discuss our results and compare them to Mokhtarian's, and also to those obtained by using the more familiar Euclidean distance-based metric.

2 Kernel-based Maximum Entropy Learning Rule

Consider a lattice A, with a regular and fixed topology, of arbitrary dimensionality d_A, in a d-dimensional input space $V \subseteq \Re^d$. To each of the N positions in the lattice corresponds a formal neuron i which possesses, in addition to the traditional weight vector $\mathbf{w}_i$, a (hyper)spherical activation region S_i, called receptive field (RF) region, with radius σ_i, in V-space (Fig. 1A). The neural activation state is represented by the code membership function:

$$\mathbb{1}_i(\mathbf{v}) = \begin{cases} 1 & \text{if } \mathbf{v} \in S_i \\ 0 & \text{if } \mathbf{v} \notin S_i, \end{cases} \tag{1}$$

with $\mathbf{v} \in V$. As the definition of S_i suggests, several neurons may be active for a given input $\mathbf{v}$. Hence, we need an alternative definition of competitive learning (Van Hulle, 1998). Define Ξ_i as the fuzzy code membership function of neuron i:

$$\Xi_i(\mathbf{v}) = \frac{\mathbb{1}_i(\mathbf{v})}{\sum_{k \in A} \mathbb{1}_k(\mathbf{v})}, \quad \forall i \in A, \tag{2}$$

so that $0 \leq \Xi_i(\mathbf{v}) \leq 1$ and $\sum_i \Xi_i(\mathbf{v}) = 1$.

With the kernel-based Maximum Entropy learning Rule (kMER), the weights $\mathbf{w}_i$ are adapted so as to produce a topology-preserving mapping; the radii σ_i are adapted so as to produce a lattice of which the neurons have an equal probability to be active (equiprobabilistic map), *i.e.*, $P(\mathbb{1}_i(\mathbf{v}) = 1) = \frac{\rho}{N}, \forall i$, with ρ a scale factor. The neuron weights $\mathbf{w}_i$ are updated as follows (Van Hulle, 1998):

$$\Delta \mathbf{w}_i = \eta \sum_{j \in A} \Lambda(i, j, \sigma_\Lambda)\, \Xi_j(\mathbf{v}^\mu)\, Sgn(\mathbf{v}^\mu - \mathbf{w}_i), \tag{3}$$

and their radii σ_i:

$$\Delta \sigma_i = \eta \left(\frac{\rho_r}{N}(1 - \mathbb{1}_i(\mathbf{v}^\mu)) - \mathbb{1}_i(\mathbf{v}^\mu) \right), \quad \forall i, \tag{4}$$

with $\rho_r \stackrel{\Delta}{=} \frac{\rho N}{N - \rho}$, $Sgn(\mathbf{x})$ the sign function acting componentwise, and $\Lambda(.)$ the usual neighborhood function, *e.g.*, a Gaussian, of which the range σ_Λ is gradually decreased during learning:

$$\sigma_\lambda = \sigma_{\lambda,0} \exp\left(-2\sigma_{\lambda,0} \frac{t}{t_{\max}} \gamma_{\mathrm{OV}} \right), \tag{5}$$

where $\sigma_{\lambda,0}$ is the initial neighborhood range, and γ_{OV} is a parameter that controls the slope of the cooling scheme ("gain"). The combined effect of the radius and weight update rules is illustrated in Fig. 1B.

3 Non-parametric Density Estimation and Optimal Smoothness

With kMER, in addition to the centers, the radii of the S_i are individually adapted such that they are activated with equal probabilities, the connection with variable kernel density estimation using radially symmetrical *Gaussian* kernels can be made easily. Hence, we obtain the following heteroscedastic Gaussian density model with equal mixtures:

$$\widehat{p_{\rho_s}}(\mathbf{v}^\mu) = \frac{1}{N} \sum_{i=1}^{N} \frac{\exp(-\frac{\|\mathbf{v}^\mu - \mathbf{w}_i\|^2}{2(\rho_s \sigma_i)^2})}{(\sqrt{2\pi}\rho_s \sigma_i)^d}, \tag{6}$$

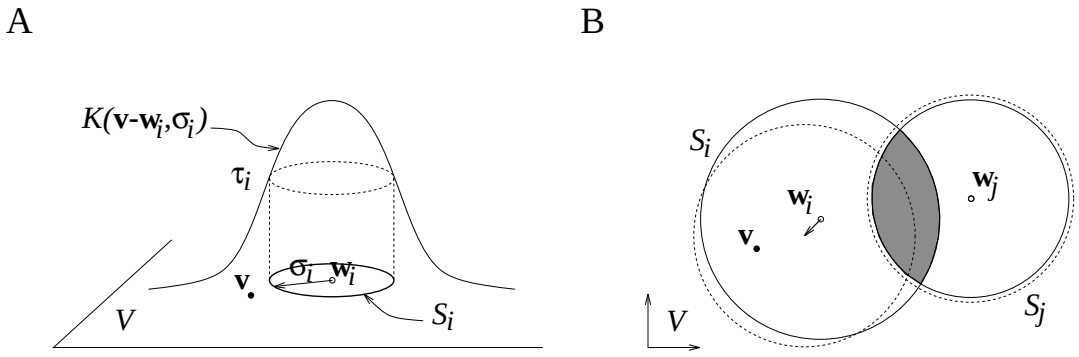

Figure 1: Kernel-based maximum entropy learning. (A) Neuron i has a localized receptive field $K(\mathbf{v}-\mathbf{w}_i, \sigma_i)$, centered at $\mathbf{w}_i$ in input space $V \subseteq \Re^d$. The intersection of K with the present threshold τ_i defines a region S_i, with radius σ_i, also in V-space. The present input $\mathbf{v} \in V$ is indicated by the black dot and falls outside S_i. (B) Receptive field region update. The arrow indicates the update of the RF center $\mathbf{w}_i$, given the present input $\mathbf{v}$ (not to scale); the dashed circles indicate the updated RF regions S_i and S_j. For clarity's sake, the range of the neighborhood function is assumed to have vanished so that $\mathbf{w}_j$ is not updated. The shaded area indicates the overlap between S_i and S_j before the update.

since the kernels are active with the same probability, with ρ_{s} a factor controlling the overall degree of smoothness. The question is now: how to choose ρ_{s}?

Contrary to the univariate case, only very few methods are available for determining the optimal smoothing factor in multivariate density estimation. For example, the method described by Luc Devroye, which has been extensively tested on univariate examples (Devroye, 1997), is not readily extendible to the multivariate case. The methods that have been described are twofold. A first class of methods is based on, or is an extension of the Least-Squares Cross-Validation (LSCV) method (for an overview, see Sain *et al.*, 1994). However, with these methods, a (Gaussian) a kernel needs to be positioned at every data point, whereas in our case, there are typically much more data points than kernels. A second class of methods adopts a binning approach, and positions a kernel at every bin's center (for an overview, see Hall and Wand, 1996). The applicability of these methods is limited to low-dimensional cases, exactly due to the required binning of the input space. Finally, one should note that both classes of methods are iterative procedures for which the optimal smoothing factors need to be derived by an exhaustive search method such as grid search.

We suggest here a new method for determining the optimal smoothing factor $\rho_{\mathrm{s,opt}}$. We locally match the data density contained in a d-dimensional hypersphere S_i, centered at $\mathbf{w}_i$,

and with radius σ_i, to the density generated by a d-dimensional Gaussian with center $\mathbf{w}_i$ and radius $\rho_s\sigma_i$. For sufficiently high input dimensions, $d > 10$, the distribution of the *distances* of the data points to $\mathbf{w}_i$ becomes approximately Gaussian, with mean $\rho_s\sigma_i\sqrt{d}$ and standard deviation $\rho_s\sigma_i/\sqrt{2}$. This can be shown as follows.

Assume a d-dimensional circularly symmetrical Gaussian with mean $[\mu_j]$, $j = 1, \ldots, d$, and standard deviation σ. The squared Euclidean distance to the center $x \triangleq \sum_{j=1}^{D}(v_j - \mu_j)^2$, $\mathbf{v} = [v_j]$, is known to obey the chi-squared distribution with $\theta = 2$ and $\alpha = \frac{d}{2}$ degrees of freedom (Weisstein, 1999):

$$p_{\chi^2}(x) = \frac{\frac{x}{\sigma^2}^{\frac{d}{2}-1} \cdot \exp\left(-\frac{x}{2\sigma^2}\right)}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})}, \tag{7}$$

for $0 \leq x < \infty$, and with $\Gamma(.)$ the gamma distribution. Hence, the Euclidean distance to the center becomes: $p(r) = 2rp_{\chi^2}(r^2)$, with $r = \sqrt{x}$, following the fundamental law of probabilities. After some algebraic manipulations, we can write the distribution of the Euclidean distances as follows:

$$p(r) = \frac{2(\frac{r}{\sigma})^{d-1}\exp\left(-\frac{r^2}{2\sigma^2}\right)}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})}. \tag{8}$$

The distribution is plotted in Fig. 2 (thick and thin continuous lines). The mean of r equals $\mu_r = \frac{\sqrt{2}\sigma\Gamma(\frac{d+1}{2})}{\Gamma(\frac{d}{2})}$, which can be approximated as $\sqrt{d}\sigma$, for d large, using the approximation for the ratio of the gamma functions by Graham and co-workers (Graham *et al.*, 1994); the second moment around zero equals $d\sigma^2$. The distribution $p(r)$ seem to quickly approach a Gaussian with mean $\sqrt{d}\sigma$ and standard deviation $\frac{\sigma}{\sqrt{2}}$ when d increases. This can be shown formally by calculating the skewness and Fisher kurtosis:

$$\text{skewness} = E(r-\mu_r)^3 = \mu_r(-2d + 1 + 2\mu_r^2), \tag{9}$$

$$\text{Fisher kurtosis} = \frac{E(r-\mu_r)^4}{E(E(r)^2 - \mu_r^2)^2} - 3 = \frac{d^2 + d(2+2\mu_r^2) - 4\mu_r^2 - 3\mu_r^4}{(d-\mu_r^2)^2} - 3. \tag{10}$$

We can verify that the skewness and Fisher kurtosis are equal to 0.116 and 3.70 10^{-2} for $d = 5$, and 8.07 10^{-2} and 8.71 10^{-3} for $d = 10$, respectively. They are plotted as a function of d in Fig. 2B. Hence, we can safely state that, for $d > 10$, the distance distribution closely resembles a Gaussian.

We now suggest to determine the optimal degree of smoothing as the one for which the mean distance to the center matches the Gaussian prediction (averaged over all neurons i):

$$\rho_{s,\text{opt}} = \frac{1}{N}\sum_{i=1}^{N}\frac{< \|\mathbf{v}^\mu - \mathbf{w}_i\| >_{\mathbf{v}^\mu \in S_i}}{\sigma_i\sqrt{d}}. \tag{11}$$

Note that, unlike traditional methods, eq. (11), does not require an iterative procedure. In order to test our method, we consider two types of input distributions: a d-dimensional

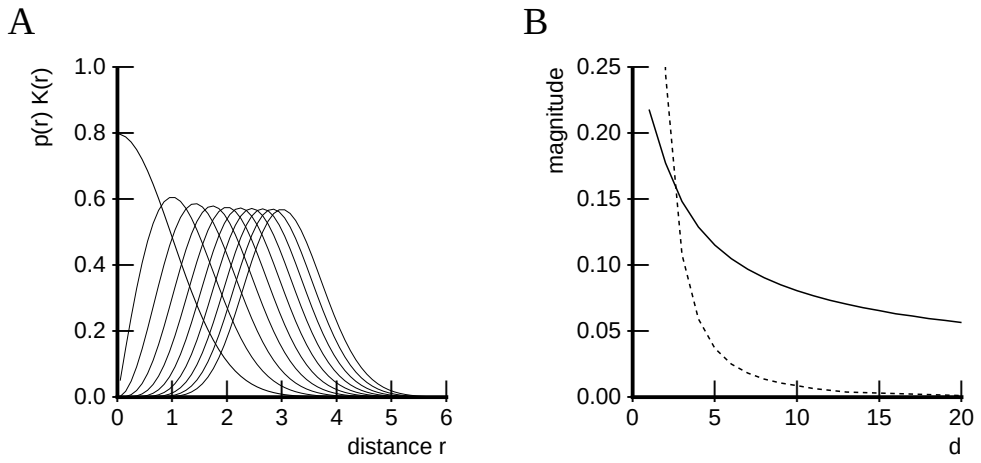

Figure 2: (A) Distribution functions of the Euclidean distance from the center of a unit-variance, radially-symmetrical Gaussian, parameterized with respect to the dimensionality d. The functions are plotted for $d = 1, \ldots, 10$ (from left to right); the thick line corresponds to the $d = 1$ case. (B) Skewness (continuous line) and Fisher kurtosis (dashed line) of the distance distributions as a function of the dimensionality d.

Gaussian $N(\mathbf{0}, 1)$, and a d-dimensional uniform distribution $[-1, 1]^d$. We train a 5×5 lattice with kMER during $t_{\max} = 2000$ time steps, with $\eta = 0.001$ and $\rho_r = 2$. The results for our method are shown in Fig. 3A and B, for the Gaussian and uniform input distributions, respectively, together with the results obtained for the LSCV method, and the theoretically optimal results which are found by minimizing the MSE between the estimated and actual *pdf*. We clearly observe that our method outperforms the LSCV method, in particular in the high-dimensional case.

4 Image Database

We will train our topographic maps on patterns taken from an image database with the purpose of detecting groups of similar images. In particular, we will consider the *Shape Queries Using Image Databases* (SQUID) (Mokhtarian *et al.*, 1996), which consists of $M = 1100$ images of contours of marine animals, with a large variety of shapes. A typical set of contours is shown in Fig. 4. One can immediately see that the contours display similarities in overall shape, tails and fins.

Since the contours are described by a varying number of points (mean 692.8 and standard deviation 205.5), we resample them to 256 points, using linear interpolation. One such result

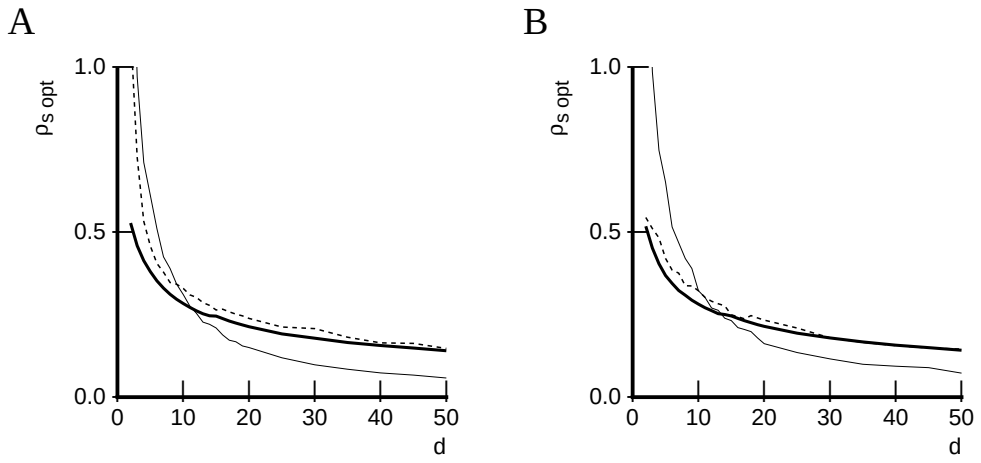

Figure 3: Optimal degrees of smoothness obtained for a Gaussian- (A) and uniform input distribution (B), as a function of input dimensionality d, using our method eq. (11) (thick continuous lines) and the LSCV method (thin continuous lines). The theoretically optimal results (dashed lines) are also shown.

is shown in Fig. 5. Afterwards, we translate every contour such that its center of mass is positioned at $(0, 0)$, and compute its Fourier transform. Only the amplitude spectrum is retained, which is further normalized so that a translation-, rotation-, and scale-invariant representation of the contour is obtained. The resulting spectra are coded by 256-dimensional vectors. An example is shown in Fig. 6. Note that this representation is also invariant to the direction in which the contour is traversed (clock- or counter-clockwise), since this only effects the phase, but not the amplitude.

The idea is now to perform a hierarchical clustering of these Fourier-transformed contours in order to detect groups of contours with similar global shapes.

5 Hierarchical Clustering Procedure

We perform a hierarchical density-based clustering analysis of $M = 1100$ normalized amplitude spectra, which have been obtained as explained in the previous section. The clustering analysis proceeds as follows. First, the probability density function (*pdf*) underlying the data is estimated with our kernel-based topographic map. Clusters correspond to high density peaks in the *pdf*-estimate and are detected without *a priori* knowledge of their number. The data set is segmented into subsets by classifying each contour to its corresponding cluster, after which the next level of the clustering analysis is performed on every such subset (hierarchical clustering).

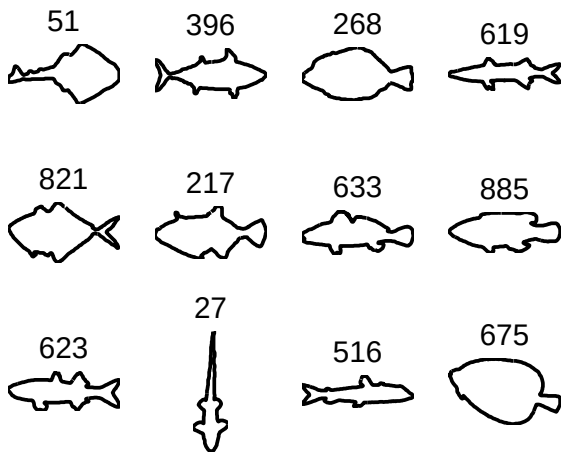

Figure 4: Examples of marine animal contours in the SQUID database. Numbers above the contours refer to the indices in the original database.

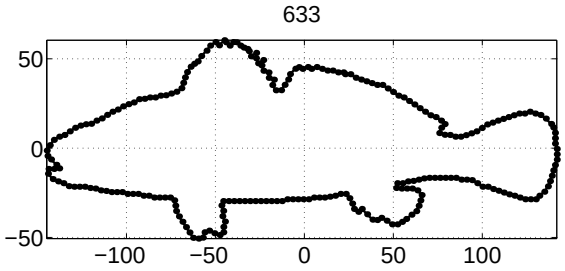

Figure 5: Normalized contour of marine animal 633, obtained by resampling and subtracting the mean of the original contour in the SQUID database.

5.1 Topographic Map Formation

We train our lattices with kMER, eqs. (3,4), using a Gaussian neighborhood function $\Lambda(.)$ and the neighborhood "cooling" scheme given in eq. (5). The initial neighborhood range $\sigma_{\lambda,0}$ is set equal to $\frac{\sqrt{N}}{2}$. The "gain" γ_{OV} in eq. (5) is optimized for, in an iterative manner, using a monitoring algorithm that tries to minimize the variability in the degree of overlap between the neurons' RF regions (Overlap Variability, OV). The lattice is more likely to be disentangled when the OV is minimal. For more details, we refer to (Van Hulle, 2000b; Van Hulle and Gautama, 2002b).

A hierarchy of topographic maps is developed with which a hierarchical clustering analysis is performed (see further). For the *root* and *Level 1* nodes in the clustering hierarchy, we use 7×7 lattices, and smaller, 5×5 lattices for the other nodes. If no clusters are detected for the smaller maps, and if the number of contours in the training set exceeds 49, we retrain

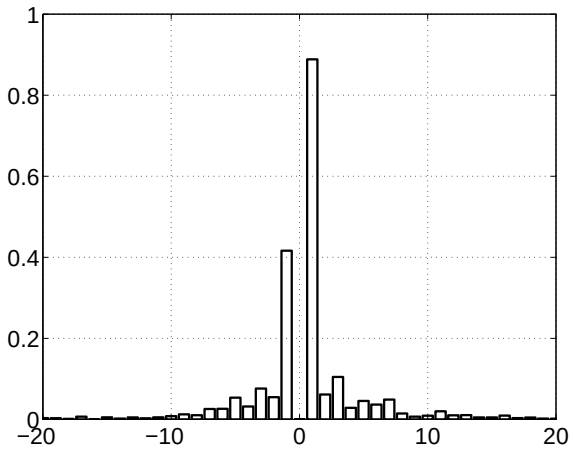

Figure 6: Normalized amplitude spectrum corresponding to Fig. 5.

the map but now using a 7×7 lattice (see Discussion).

5.2 Density Estimation

We construct a kernel-based density estimate for each topographic map developed in the previous stage, and determine the optimal smoothing factor $\rho_{\mathrm{s,opt}}$, as explained in Section 3.

5.3 Clustering and Labeling the Topographic Map

We apply the discrete hill-climbing algorithm described in (Van Hulle, 2000a,b). The algorithm determines the number c, and the location of the local density peaks in the *pdf* estimate eq. (6), which is evaluated only at the neurons' weights, that are not surmounted by higher peaks in a range of k nearest neurons. The number of clusters is determined and plotted as a function of k, in the valid range $k = 1 \ldots \frac{N}{2}$. As the final number of clusters, we take the number that corresponds to the longest plateau in the plot. The plateau for $c = 1$ is rejected, since this is a degenerate case. Figure 7A shows an example plot, namely that found for the *root* node of the cluster hierarchy. Finally, the neurons in the lattice are labeled according to which local density peak they belong. For this we take the clustering result for k at the beginning of the plateau. Each neuron receives a greyscale according to which cluster it belongs to. This results in a *cluster map*, an example of which is shown in Fig. 7B, for the *root* node in the cluster hierarchy.

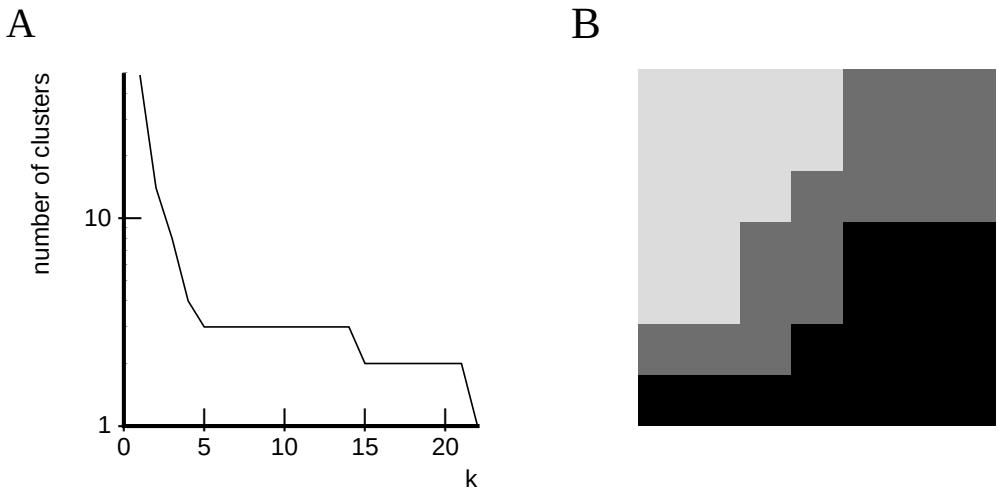

Figure 7: (A) Number of clusters that are detected as a function of the number of nearest neurons, k, in the discrete hill-climbing algorithm for the *root* node. The longest plateau is found for three clusters, starting at $k = 5$. (B) Cluster map for the *root* node corresponding to $k = 5$.

5.4 Labeling the Training Set

We now need a metric for assigning input patters (*i.e.*, contour amplitude spectra) to clusters. We opt for one based on outlier detection, since outlier probabilities are one-dimensional quantities. To each cluster corresponds a (cluster-conditional) *pdf* estimate. An input pattern is classified to the cluster for which the probability of being an outlier is minimal (*i.e.*, an outlier with respect to the cluster's *pdf* estimate). There is, however, a problem due to the high dimensionality of our data: the Gaussian kernels cannot be evaluated directly, since this leads to numerical instabilities (see the factor σ_i^d in eq. (6)). We have, therefore, developed the following new method.

Rather than evaluating the high-dimensional Gaussians directly, we evaluate the one-dimensional distributions that describe the distances to the kernel centers, which are approximately Gaussian for $d > 10$, with means $\rho_s \sigma_i \sqrt{d}$ and standard deviation $\frac{\rho_s \sigma_i}{\sqrt{2}}$ (see section 3). By observing the corresponding cumulative distribution, we compute the cluster outlier probability for cluster n, p_n, using:

$$p_n(\mathbf{v}^\mu) = \frac{1}{2N_n} \sum_{i \in \text{cluster } n} \left(1 + \text{erf} \left(\frac{\|\mathbf{v}^\mu - \mathbf{w}_i\| - \rho_{s,\text{opt}} \sigma_i \sqrt{d}}{\rho_{s,\text{opt}} \sigma_i} \right) \right), \tag{12}$$

where N_n is the number of neurons that belong to cluster n. Input pattern $\mathbf{v}^\mu$ is classified to the cluster for which the outlier probability, p_n, is the smallest. We apply this method, rather than a straightforward computation of the cluster probability, since the latter yields

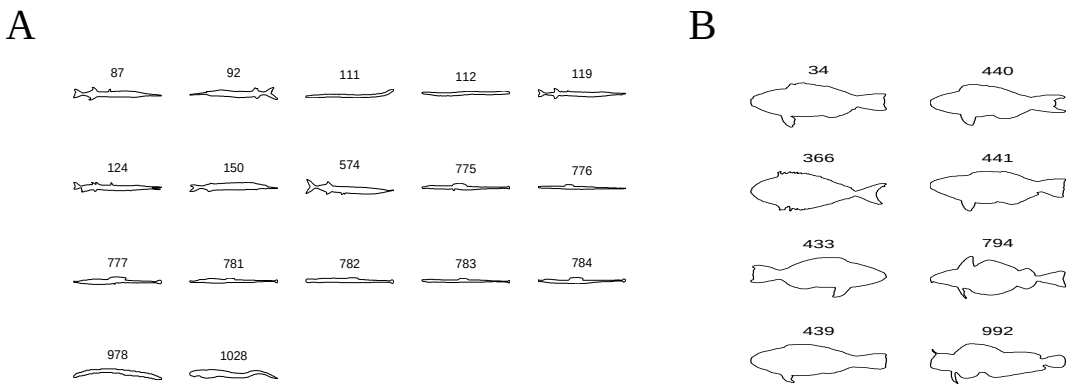

Figure 8: Contours that are classified to a *Level 2* cluster ([2 1]) (A), and a *Level 7* cluster ([0010012]) (B). The numbers above the contours refer to the indices in the original database.

numerically unstable results for higher dimensions due to the normalization term for unit-volume Gaussians. Furthermore, it offers the advantage over a Euclidean distance-based one, where an input pattern receives the same label as its nearest neuron (as is used in Gautama and Van Hulle, 2000), since it takes into account the width of the clusters. However, since in high-dimensional spaces the erf function is steep, the case where $p_n = 1$ for all clusters can occur and, hence, a tie-breaking strategy is needed. We opt for the smallest Euclidean distance to the nearest neuron as a tie-break.

5.5 Hierarchical Clustering

The set of input patterns on which a given node in the hierarchy is trained is segmented into subsets according to the patterns's classification labels obtained in the previous step. The clustering analysis is then re-applied to each subset. The procedure terminates, and produces a leaf node in the clustering hierarchy, if the discrete hill-climbing algorithm does not yield a plateau, or if the number of input patterns applied to a given node does not exceed the number of neurons in the node's topographic map.

In total, we have detected 65 clusters corresponding to leaf nodes in the clustering hierarchy (the deepest leaf node is situated at *Level 10*, with the *root* node being at *Level 0*). The first cluster is found at *Level 2* and is shown in Fig. 8. Possibly, this cluster is found at an early level because: 1) the overall shape is very different from the others, and 2) there are sufficient *similar* contours in the database to define a cluster at this level.

Note that the labeling of the data occurs in a hierarchical manner: every data point "traverses" the tree until it arrives at a leaf node. An alternative method would be to construct a partial *pdf* estimate for every leaf node in the hierarchy. If every cluster is considered

equiprobable, a contour could be classified to the corresponding *pdf* estimate for which the outlier probability eq. (12) is the smallest. This would yield different results for data points that lie in the border region between clusters: the *pdf* estimates become more detailed at deeper levels, due to which the boundaries can shift slightly.

6 Discussion

6.1 Shape Clustering

In the literature, several clustering-based procedures are described for constructing representation structures for sets of contours (Dubuission *et al.*, 1996; Lee and Street, 2000). These procedures assume prior knowledge of the number of clusters in the dataset. In a different context, namely that of querying large databases for contours that are similar to the target, various shape similarity metrics have been suggested (Mokhtarian, 1995; Mokhtarian *et al.*, 1996; Niblack *et al.*, 1993). We have introduced in this article a novel way of constructing a representation structure for a set of contours, which can be used for shape classification purposes. The advantage is that it uses a data *model* which, in general, greatly facilitates data handling. Our procedure relies on a hierarchical, density-based clustering approach.

6.2 Shape Clustering with Our Approach

There are two aspects about the shapes in the SQUID-database that make it hard to cluster them. First, there is the wide variety in shapes, due to which one would expect a lot of clusters to be detected. Second, the number of patterns that defines a single cluster is very small, which makes kMER-training more difficult (few training patterns and high dimensionality). For example, sometimes no clusters were detected in relatively large subsets ($M > 49$ input patterns), where clusters would be expected. In these cases, the estimated *pdf* is not detailed enough and is too smooth to indicate the presence of clusters. Indeed, since neuron i is activated by $M_i = \frac{\rho_r M}{N - \rho_r}$ samples in the kMER-algorithm (Van Hulle, 1998), several clusters are modeled by a single kernel if the number of patterns belonging to one cluster (which is not known *a priori*) is less than M_i. Thus, the resulting kernels are too wide to yield distinct local peaks in the *pdf*-estimate. Therefore, we retrain the kMER-algorithm using a 7×7 lattice if no clusters are detected at a given node, and only terminate the analysis for that node if the latter map does not yield any clusters either. In this way, a more detailed *pdf* estimate can be developed if necessary. This was the case for 4 nodes in our clustering hierarchy. The converse situation has not posed any problems: the kMER-algorithm and the

subsequent *pdf* estimation has proven to be robust even when there are very few contours in the training set: with as few as 25 contours, we have trained 5×5 topographic maps, that still showed the presence of clusters.

As a measure of confidence with which a given contour is classified at a certain level in the hierarchy, we compute the difference between the lowest and second-lowest outlier probability taken over the subclusters of the same parent cluster (*i.e.*, "sister"-clusters), which will be equal to 1 if the contour is classified with 100% certainty, and 0 if both probabilities are the same, possibly when the contour is a definite outlier for both clusters and where the Euclidean distance metric has been used as a tie-break. We have computed this for the 65 leaf nodes in the clustering hierarchy. The mean classification confidence is 0.121. This is a relatively small number, possibly due to the high dimensionality of the data, $d = 256$, and the relatively small number of important dimensions: a principal component analysis shows that the cumulative eigenvalue plot reaches 99% for 20 principal components. This means that most of the variance is concentrated in a subspace, which has its implications on the computation of the outlier probability, eq. (12), and possibly even on the computation of the optimal degree of smoothing, eq. (11).

Figure 9 shows two example clusters (*Level 4* in Fig. 9A and *Level 5* in Fig. 9B). For one contour in every cluster, which would intuitively be considered an outlier (the framed ones in Fig. 9), the classification confidences are computed. The classification confidence for contour 780 in Fig. 9A is 6.5×10^{-6} (the mean confidence for this cluster is 0.121). The Euclidean distances to the nearest neurons in the best and second-best clusters are 0.110 and 0.212, respectively (mean distances between patterns and nearest neurons is 0.050). The large distances indicate that these contours are indeed located "in between" the sister clusters. The second case (Fig. 9B) shows a similar situation. Contour 543 (in cluster [0 0 1 0 1]) has a zero classification confidence (the Euclidean distance metric has been used as a tie-break), and is, thus, an outlier to all "sister"-clusters, *i.e.*, those coming from [0 0 1 0]. The mean classification confidence for this cluster is 0.077. The Euclidean distances between this contour and the nearest neurons in the two sister clusters are 0.174 and 0.228 (the mean distance between patterns belonging to this cluster and the nearest neuron is 0.068).

6.3 Comparison with Mokhtarian's Approach

Although there is no objective way of quantifying the clustering performance, since the database is unlabeled, we compare our results to those obtained with the Curvature Scale Space (CSS) retrieval system developed by Mokhtarian and co-workers (Mokhtarian *et al.*,

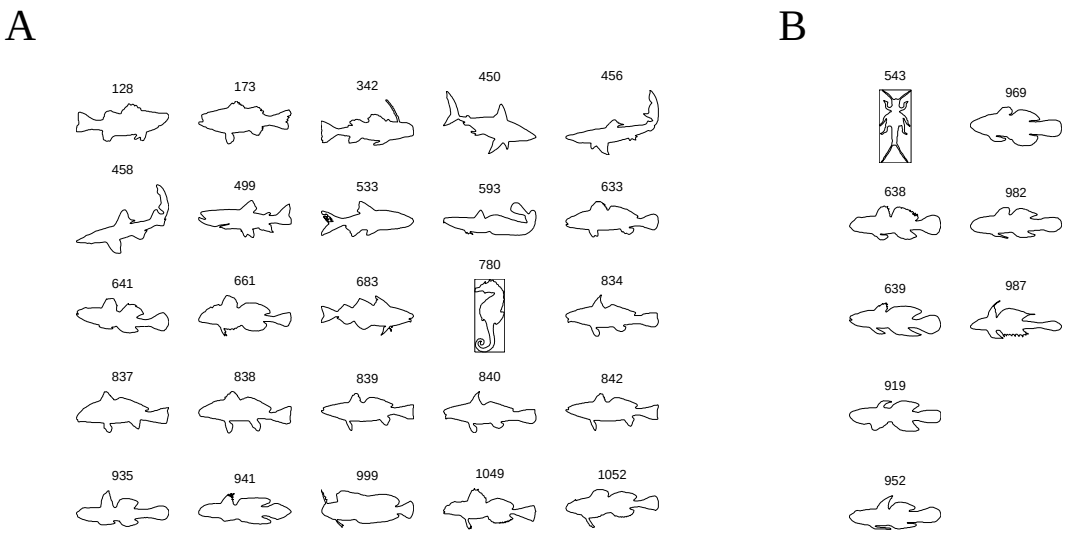

Figure 9: Contours that are classified to a *Level 4* cluster [1 0 1 2] (A), and a *Level 5* cluster [0 0 1 0 1] (B). The target contours are indicated by a rectangle.

1996). Examples of their results are shown in Fig. 10A and D: the upper left contour corresponds to the "target", and the results are ordered by their measure of similarity, downwards, then rightwards. The clusters that we find, and that contain the target contour, are shown in Fig. 10B and E, respectively. In the first example (Fig. 10A and B), there is a good correspondence between the two sets: all but one contour that have been found by the CSS retrieval system are contained in the corresponding cluster of our analysis. However, in the second case, the two sets are disjoint (except for the target contour). This is due to the difference in similarity criteria. The CSS retrieval system takes into account only the positions of the topological landmarks (with zero-curvature), and not so much the fine shape details. The set in Fig. 10D, *e.g.*, possibly groups contours with seven points with zero-curvature (two for the tail, four fins and one for the snout), whereas the Fourier representation we use takes into account the global shape. However, we do not claim that the Fourier descriptor is better suited for representing shapes than the topological landmark-based descriptor used by the CSS retrieval system.

6.4 Comparison with Euclidean Distance-based Approach

Finally, we can also perform our clustering analysis using a criterion based on Euclidean distance, instead of outlier detection. This results in a different clustering hierarchy (only 53 nodes, with the deepest node in *Level 7*). The clusters for target contours 262 and 102 are shown in Fig. 10C and F, respectively. It seems that the results are now inferior, since

both clusters contain mixtures of shapes. However, again, since the database is unlabeled, we cannot objectively say which method yields the best results.

7 Conclusion

We have introduced a new way to optimally smooth, by a common scale factor, heteroscedastic Gaussian density models with equal mixtures. In addition, we have introduced a new way to build a representation structure for images of shapes, based on the Fourier representation of their contours, and a new method for labeling samples, based on outlier detection. All this has been incorporated into a hierarchical clustering procedure that relies on kernel-based topographic maps. The clustering procedure has been applied to the SQUID image database (Mokhtarian *et al.*, 1996). The advantage of our approach is that the number of shape clusters is obtained *without* prior knowledge, and that a hierarchy of shapes is generated. Finally, we have compared our clustering results to those obtained with the CSS retrieval system developed by Mokhtarian and co-workers (Mokhtarian, 1995; Mokhtarian *et al.*, 1996), as well as to those obtained when using the Euclidean distance metric.

Acknowledgments

The authors would like to thank Dr. Farzin Mokhtarian, University of Surrey (UK), for making the SQUID-database available. M.M.V.H. is supported by research grants received from the Fund for Scientific Research (G.0185.96N), the National Lottery (Belgium) (9.0185.96), the Flemish Regional Ministry of Education (Belgium) (GOA 95/99-06; 2000/11), the Flemish Ministry for Science and Technology (VIS/98/012), and the European Commission (QLG3-CT-2000-30161 and IST-2001-32114). T.G. is supported by a scholarship from the Flemish Regional Ministry of Education (GOA 2000/11).

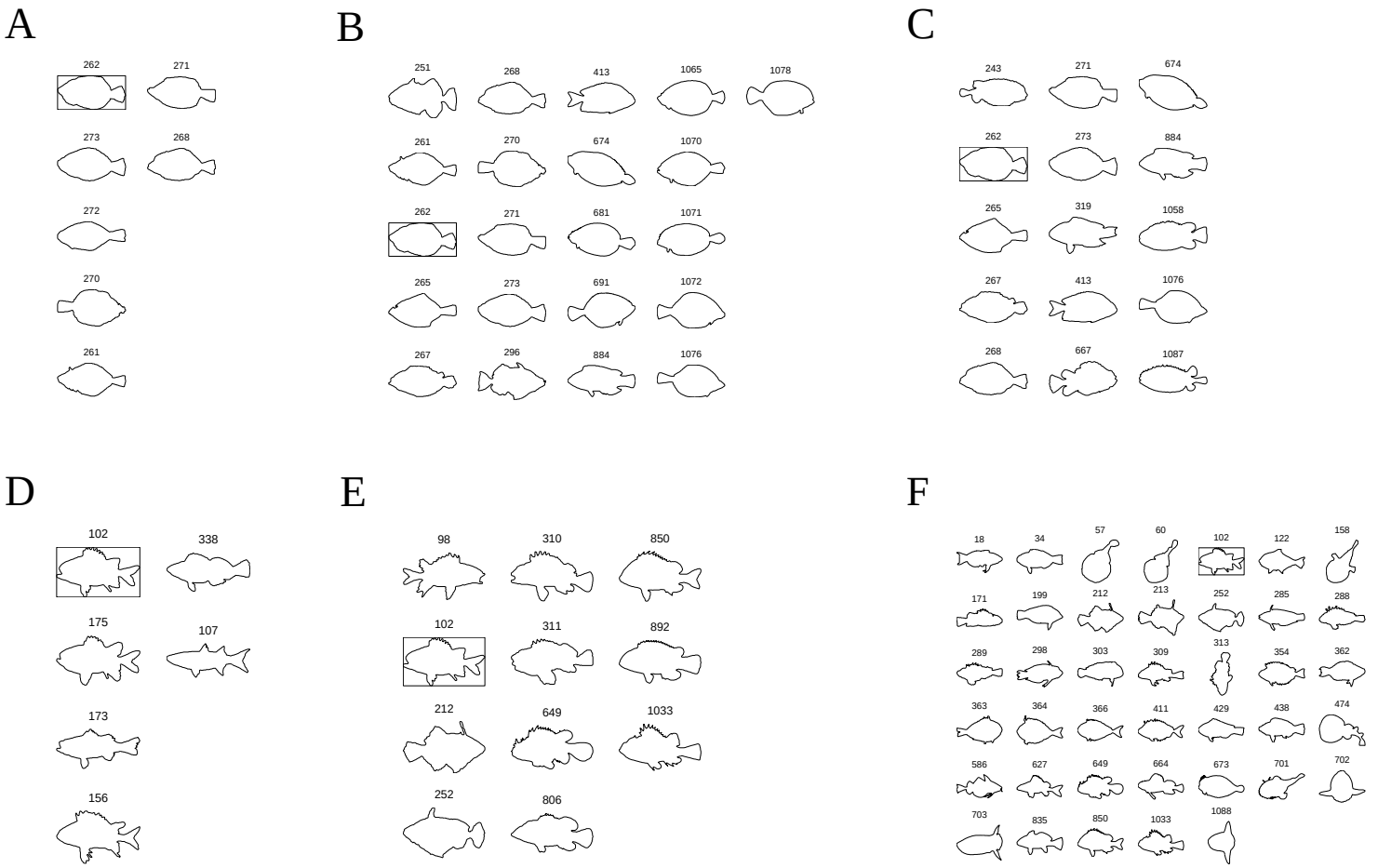

Figure 10: The contours that best match the target contours 262 and 102 (framed contours), obtained with three different methods (columns). A,D) Best matching results obtained with the CSS retrieval system of Mokhtarian and co-workers, when querying the database with the upper left contour (indices 262 and 102, redrawn from Fig. 4a and 4b in Mokhtarian *et al.*, 1996). The contours are ordered by decreasing similarity, downwards, then rightwards. B,E) The corresponding clusters found with our hierarchical clustering procedure. C,F) The clusters found with the Euclidean distance metric.

References

Cottrell, M., Gaubert, P., Letremy, P., and Rousset, P. (1999). Analyzing and representing multidimensional quantitative and qualitative data: Demographic study of the Rhône valley. The domestic consumption of the Canadian families. In *Kohonen Maps* (Proc. WSOM99, Helsinki), E. Oja and S. Kaski (Eds.), pp. 1–14.

Deboeck, G.J., and Kohonen, T. (1998). *Visual explorations in finance with self-organizing maps.* Heidelberg: Springer.

Dersch, D.R., and Tavan, P. (1995). Asymptotic level density in topological feature maps. *IEEE Trans. Neural Networks*, **6**, 230–236.

Devroye, L. (1997). Universal smoothing factor selection in density estimation: theory and practice (with discussion), *Test*, **6**, 223–320.

Dubuission Jolly, M.P., Lakshmanan, S., and Jain, A.K. (1996). Vehicle Segmentation and Classification Using Deformable Templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **18**(3), 293–308.

Gautama, T., and Van Hulle, M.M. (2000). Hierarchical Density-based Clustering In High-dimensional Spaces Using Topographic Maps. *IEEE Neural Network for Signal Processing Workshop 2000, Sydney*, 251–260.

Graham, R.L., Knuth, D.E., and Patashnik, O. (1994). Answer to problem 9.60 in *Concrete Mathematics: A Foundation for Computer Science.* Reading, MA: Addison-Wesley.

Hall, P., and Wand, M.P. (1996). On the accuracy of binned kernel density estimators. *Journal of Multivariate Analysis*, **56**, 165–184.

Himberg, J., Ahola, J., Alhoniemi, E., Vesanto, J., and Simula, O. (2001). The Self-Organizing Map as a Tool in Knowledge Engineering. In: *Pattern Recognition in Soft Computing Paradigm*, Nikhil R. Pal (ed.), pp. 38–65, World Scientific Publishing: Singapore.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.*, **43**, 59–69.

Kohonen, T. (1984). *Self-organization and associative memory.* Heidelberg: Springer.

Kohonen, T. (1995). *Self-organizing maps.* Heidelberg: Springer.

Krishnaiah, P.R., and Kanal, L.N. (1982). *Classification, Pattern Recognition, and Reduction of Dimensionality*, Handbook of Statistics, Vol. 2, Amsterdam: North Holland.

Lagus, K., and Kaski, S. (1999). Keyword selection method for characterizing text document maps. *Proc. ICANN99, 9th Int. Conf. on Artificial Neural Networks*, IEE: London, Vol. 1, pp. 371–376.

Lee, K.-M., and Street, N. (2000). Automatic Image Segmentation and Classification Using On-line Shape Learning. *Fifth IEEE Workshop on the Application of Computer Vision, Palm Springs, CA, USA.*

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Stat. and Prob.*, (Vol. 1), pp. 281–296.

Mokhtarian, F. (1995). Silhouette-based isolated object recognition through curvature scale space. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**(5), 539–544.

Mokhtarian, F., Abbasi, S., Kittler, J. (1996). Efficient and Robust Retrieval by Shape Content through Curvature Scale Space. *Proc. International Workshop on Image DataBases and MultiMedia Search, Amsterdam, The Netherlands*, pp. 35–42.

Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., and Yanker, P. (1993). The QBIC Project: Querying Images By Content Using Color, Texture, and Shape. *SPIE*, **1908**, 173–187.

Ritter, H. (1991). Asymptotic level density for a class of vector quantization processes. *IEEE Trans. Neural Networks*, **2**(1), 173–175.

Sain, S.R., Baggerly, K.A., and Scott, D.W. (1994). Cross-validation of multivariate densities. *Journal of the American Statistical Association*, **89**(427), 807–817.

Van Hulle, M.M. (1998). Kernel-based equiprobabilistic topographic map formation. *Neural Computation*, **10**, 1847–1871.

Van Hulle, M.M. (2000a). *Faithful Representations and Topographic Maps. From Distortion- to Information-Based Self-Organization* (Haykin, S. ed.), New York: Wiley.

Van Hulle, M.M. (2000b). Monitoring the Formation of Kernel-based Topographic Maps. *IEEE Neural Network for Signal Processing Workshop 2000, Sydney*, 241–250.

Van Hulle, M.M. (2002a). Kernel-based topographic map formation by local density modeling. *Neural Computation*, in press.

Van Hulle, M.M., and Gautama, T. (2002b). Monitoring the formation of kernel-based topographic maps with application to hierarchical clustering of music signals. *J. VLSI Signal Processing Systems for Signal, Image, and Video Technology*, in press.

Vesanto, J. (1999). SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, **3**(2), 111–126.

Vesanto, J., and Alhoniemi, E. (2000). Clustering of the Self-Organizing Map. *IEEE Trans. Neural Networks*, **11**(3), 586–600.

Weisstein, E.W. (1999). *CRC Concise Encyclopedia of Mathematics.* London: Chapman and Hall.

Brief Author Biographies

Marc M. Van Hulle received a M.Sc. in Electrotechnical Engineering (Electronics) and a Ph.D. in Applied Sciences from the K.U.Leuven, Leuven (Belgium) in 1985 and 1990, respectively. He also holds B.Sc.Econ. and MBA degrees. In 1992, he has been with the Brain and Cognitive Sciences department of the Massachusetts Institute of Technology (MIT), Boston (USA), as a postdoctoral scientist. He is affiliated with the Neuro- and Psychophysiology Laboratory, Medical School, K.U.Leuven, as an associate Professor. He has authored the monograph *Faithful representations and topographic maps: From distortion- to information-based self-organization*, John Wiley, 2000, which is also translated in Japanese, and more than 80 technical publications. (http://simone.neuro.kuleuven.ac.be)

Dr. Van Hulle is an Executive Member of the IEEE Signal Processing Society, Neural Networks for Signal Processing (NNSP) Technical Committee (1996-2003), the Publicity Chair of NNSP's 1999, 2000 and 2002 workshops, and the Program co-chair of NNSP's 2001 workshop, and reviewer and co-editor of several special issues for several neural network and signal processing journals. He is also founder and director of Synes N.V., the data mining spin-off of the K.U.Leuven (http://www.synes.com). His research interests include neural networks, biological modeling, vision, data mining and signal processing.

Temujin Gautama received a B.Sc. degree in electronical engineering from Groep T, Leuven, Belgium and a Master's degree in Artificial Intelligence from the Katholieke Universiteit Leuven, Belgium. He is currently with the Laboratorium voor Neuro- en Psychofysiologie at the Medical School of the K.U.Leuven, where he is working towards his Ph.D.
His research interests include nonlinear signal processing, biological modeling, self-organizing neural networks and their application to data mining.