

# CONSIDERING TOPOLOGY IN THE CLUSTERING OF SELF-ORGANIZING MAPS<sup>1</sup>

**Kadim Taşdemir, Erzsébet Merényi**

Rice University

Houston, Texas, USA

{tasdemir, erzsebet}@rice.edu

**Abstract** – *The Self-Organizing Map (SOM) [1] is an effective tool for clustering and data mining. One way to extract cluster structure from a trained SOM is by clustering its weights, which has great potential for automation. This potential is not fully realized by existing algorithms, and leaves large, high-dimensional, complex data to semi-manual treatment. Our main contribution is the exploitation of the data topology in clustering the SOM. Combined with appropriate distance and cluster validity measures, this results in a high degree of precision and automation of cluster extraction, including the discovery of rare clusters. It may work with prototypes of other quantization methods since direct use of SOM locations can be avoided.*

**Key words** – SOM, clustering, boundary extraction, data mining

## 1 Introduction

The Self-Organizing Map (SOM) [1] is a widely and successfully used neural paradigm for clustering. Automated and precise capture of cluster boundaries from a learned SOM has been a long-standing challenge that to date has only partial solutions. The problem is especially important for high-dimensional and large data sets with many meaningful clusters such as in remote sensing or medical imagery, which often also have interesting rare clusters to be discovered. The U-matrix approach [2] and its variants work well when relatively large SOM grid is applied to small data sets with a low number of clusters (e.g., [3], [4], [5]) but they tend to obscure finer structure for more complex data due to averaging of weight distances over neighbours or thresholding. Imaginative approaches such as [6] and gravitational methods (e.g., Adaptive Coordinates [4]) visualize distances between receptive field centres in informative ways that greatly guide the human operator. However, they do not extract clusters explicitly. Experiments with automated colour assignments are admittedly meant for qualitative exploration of the approximate cluster structure [7], [8], [9]. More overview and references are offered in [4]. [10] presents a growing SOM that handles high data dimension gracefully, but it appears less robust than the Kohonen SOM (KSOM) because of the larger number of parameters to adjust, and it is unclear how it would work for large data volumes. An elegant proposal is the use of higher order neurons [11] whereby clusters of highly complex shape can be represented by single tensorial weights. This is theoretically very attractive but has several problems, one being the computational complexity; another is the need for tight determination of the number of SOM neurons because of the one-to-one correspondence

---

<sup>1</sup> This work was partially supported by grant NNG05GA94G from the Applied Information Systems Research Program, NASA, Science Mission Directorate. Figures are in color, request color copy by email.

between weights and clusters. Two-step algorithms cluster the SOM weights and then assign each data point into the cluster of the prototype vector that they map to, which provides an automated cluster extraction. However, there are many details that determine the quality of the resulting clustering, especially for complicated, high-dimensional real data. [12] uses a hierarchical classification based on Ward measure to get superclusters from the SOM weights. [13] provides a detailed discussion to which we refer the reader due to space constraints. Common to the above works is a low number of clusters and a modest number of data points. Some use the size of the receptive fields of the weights in cluster validity criteria or for visualization (e.g., [5], [8]), but none exploit the topology. [14] presents a limited use of the SOM lattice connections, and applies successfully to large real data. We contribute to by taking into account the topology, besides the local density, in the clustering. Our algorithm works in a highly automated fashion for large real data with many clusters of widely varying statistics including small rare clusters.

## 2 Clustering of the SOM

Clustering data with an SOM consists of two steps. First the SOM is trained with any of several available learning rules [1] so that the weights of the Processing Elements (PEs, neural units), become optimally placed prototype vectors of the data space. After training, groups of similar weights are identified and data points assigned to the cluster of their prototype vector. This can be done in a number of ways including the methods referred to in the Introduction. Clustering the SOM weights is of particular interest because – if high quality – its potentials to both find detailed structure and provide a high level of automation are great. We propose a hierarchical agglomerative scheme, similar in its main steps to that in [13]. However, we introduce substantial differences, which we discuss at the respective points below. Some of these differences are adapted from [15], which we regard as the precursor of this work. We refer to [13] for a discussion of the relative merits of partitive and hierarchical approaches, and only recall here that the latter can handle arbitrary cluster shapes with appropriate metrics and it is of relatively low complexity due to the use of local information. It suits high-dimensional real data, which often are hard to describe with parametric models. We aim at detailed cluster identification of voluminous high-dimensional data by including two components of the SOM knowledge that were underutilized in previous methods: topology and local data density. We show, in particular, that this facilitates the capture of rare clusters.

### 2.1 Our Proposed Algorithm

Defining characteristics of all clustering methods are the between- and within-cluster metrics and the cluster validity criterion used. Table 1 lists the metrics used in this paper. Our cluster validity criterion is the same as in [15]: *A point in a cluster is closer to some point in the same cluster than to any point in any other cluster.* This is in contrast to [13], where the cluster validity measure is the Davies-Bouldin index or a gap criterion. We follow [15] because it facilitates identification of finer cluster structure. The between- and within-cluster distances are single linkage and maximum nearest neighbour, respectively, in [15] as well as here. [13] uses centroid linkage and centroid distance; or single linkage and nearest neighbour respectively, for the same. We cluster the SOM in four Phases, following the general logic of [13].

**Phase I. Building a dendrogram:** Each weight vector is a singleton cluster at first. Pairwise single linkage distances are computed between all clusters and the two clusters closest to each other are merged. This process is repeated until only one cluster is left. PEs with empty receptive fields are interpolating units in the SOM. For complex or noisy data, non-empty PEs can still be interpolating units, characterized by small receptive fields. Omitting these from the

**Table 1.** Metrics for within-cluster and between-clusters distances.  $C_k$ , and  $C_l$  represent clusters.  $d_w(C_k)$  is the within-cluster distance of  $C_k$ ,  $d_b(C_k, C_l)$  is the distance between clusters  $C_k, C_l$  where  $x_i, x_j \in C_k$  and  $y_j \in C_l$  are weights (points),  $c_k, c_l$  are the centres in the corresponding clusters.  $N_k$  is the number of weights in  $C_k$ .

Within-cluster distance $d_w(C_k)$	Between-cluster distance $d_b(C_k, C_l)$
Nearest neighbour $d_{w\_nn} = \sum_i \min_j \ x_i - x_j\  / N_k$	Complete linkage $d_{b\_comp} = \max_{i,j} \ x_i - y_j\ $
Max. nearest neighbour $d_{w\_nn\_max} = \max_i \min_{j:j \neq i} \ x_i - x_j\ $	Single linkage $d_{b\_slink} = \min_{i,j} \ x_i - y_j\ $
Centroid distance $d_{w\_cen} = \sum_i \ x_i - c_k\  / N_k$	Centroid linkage $d_{b\_cen} = \ c_k - c_l\ $

tree construction greatly improves cluster separation. In [13], all non-empty weights are included. Here, we include the weights whose receptive field size,  $|RF_i|$ , satisfies

$$|RF_i| > \mu_{RF} - \alpha \cdot \sigma_{RF} \quad \alpha > 0 \text{ scalar control parameter} \quad (1)$$

where  $\mu_{RF}$  is the mean of  $|RF_i|$  over *all* SOM units,  $\sigma_{RF}$  is the standard deviation. To omit at least the empty weights,  $\alpha \leq \mu_{RF} / \sigma_{RF}$  must hold. A smaller  $\alpha$  can help with noisy data since there will be more interpolating units in that case. For complex data, larger  $\alpha$  will help include more weights, for better representation of the cluster structure.  $\alpha$  should be large enough to include most of the weights. We decide about the weights that do not satisfy eq. (1), in Phase IV.

**Phase II. Determining potential merges:** A node in the dendrogram may or may not represent a separate cluster. Therefore pruning is needed to find potential merges. [13] used the Davies-Bouldin index and a gap criterion for this purpose. The Davies-Bouldin index calculates the separation degree between clusters based on centroid distances ( $d_{b\_cen}, d_{w\_cen}$ ). This assumes spherical clusters, which is usually not valid for real data. The gap criterion compares the single linkage distance between clusters to the sum of the nearest neighbour within-cluster distances. Singleton clusters are not allowed in [13], which may result in missing rare clusters. To retain the discovery potential we use maximum nearest neighbour distance for within-cluster, and single linkage for between-cluster metric. These metrics can be sensitive to noise and outliers in direct clustering of the data but less so in clustering the SOM weights. Omitting the interpolating weights further reduces the undesirable effects of noise. As illustrated in Figure 1, a node C is labeled for *potential merge* if the maximum of the nearest neighbour distances within its children A, B, and the single linkage between them meet the criterion set in [15]:

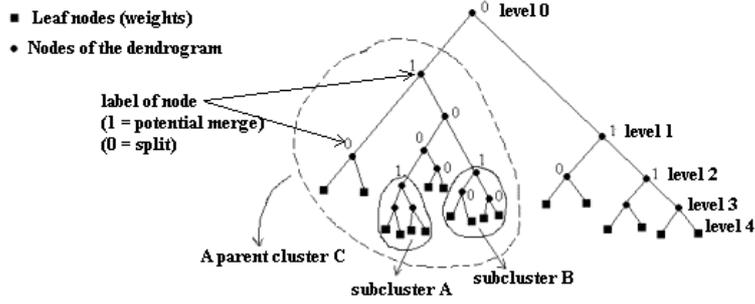
$$\frac{\max(d_{w\_nn\_max}(A), d_{w\_nn\_max}(B))}{d_{b\_slink}(A, B)} > \frac{level + 1}{N - 1} \quad (2)$$

$N$  is the number of weights used in the tree construction, *level* is the level in the tree. Scaling the RHS with *level* increases the chance of a node being a potential merge, especially near the top where groups of larger variance are more probable. Yet it helps obtain finer cluster structure because clusters labelled as potential merges will be eligible for partitioning in Phase III.

**Phase III. Partitioning:** Potential merges are examined to decide whether to keep as a cluster or to partition, according to the following criterion and as illustrated in Figure 1.

$$\sum_{i \in A} |RF_i| + \sum_{j \in B} |RF_j| > \sum_{k \in (C - (A \cup B))} |RF_k| \quad (3)$$

The criterion in [13] is to partition if the number of weights in the subclusters is greater than the number of loose leaves. We replace the number of weights with  $|RF_i|$  of those weights to represent the data density, as in [15]. This improves the chance of finding singleton clusters.



**Figure 1.** Partitioning illustration (Phase III). A, B, and C are potential clusters in the left part of the tree. The parent C contains two subclusters, A and B, with a total of 8 leaves, and 4 loose leaves. If the size of the combined receptive fields of the weights in A and B is smaller than that of the loose leaves, C is kept as a cluster.

**Phase IV. Final merging of clusters:** Clusters created in Phases I – III are checked if, according to our validity measure, they should be merged. Because of the level based determination of potential merges in Phase II, we will have more clusters at this point than [13]. In [13] each singleton cluster is merged into the cluster closest to it. We consider the following cases, similar to [15]: 1) merge of two non-singleton clusters; 2) merge involving one or two singleton clusters. A parameter to control granularity is introduced in [15], however, it proved difficult to set for different data sets. Our approach is to use the data topology instead, in the decision about merges. We define a *connectivity strength matrix*, *CONN*, as follows:

**Definition:** Let *CONN* be an  $M \times M$  matrix where  $M$  is the number of all SOM PEs. The connectivity strength,  $CONN(i,j)$ , of two PEs  $i$  and  $j$ , is the number of data samples for which  $i$  or  $j$  is the best matching unit (BMU) and the other is the second BMU.

*CONN* is a weighted analogue of the adjacency matrix obtained by Delunay triangulation, symmetric by definition, showing how strongly adjacent weights are connected in data space. Since SOM locations of the weights are not involved, topology violations do not invalidate its use.

Case IV/1) Two non-singleton clusters,  $C_k$  and  $C_l$ , are merged if the maximum nearest neighbour distance within them is larger than their between-cluster distance (similarly to eq. (2) but with a 1 on the RHS; and similarly to criteria in [13] but with different distance metrics).

$$\frac{\max(d_{w\_nn\_max}(C_k), d_{w\_nn\_max}(C_l))}{d_{b\_slink}(C_k, C_l)} > 1 \quad (4)$$

Case IV/2) If one or both of  $C_k$  and  $C_l$  are singleton, they are merged if either they satisfy eq. (4), or their connectivity strength is above a threshold (eq. (5)). The connectivity strength of clusters  $C_k$  and  $C_l$  is the connectivity strength of the weights  $w_i \in C_k$ , and  $w_j \in C_l$  that form the single linkage distance between them.

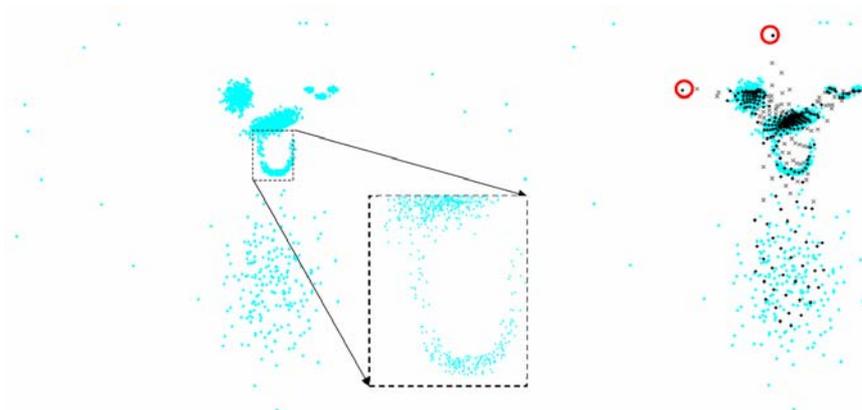
$$CONN(i, j) > \gamma \left( \frac{|RF_i|}{n_i} + \frac{|RF_j|}{n_j} \right) \quad (5)$$

$n_i$  and  $n_j$  are the number of neighbours of  $i$  and  $j$ , respectively, in the SOM lattice.  $n_i$  and  $n_j$  are the same except for borders units.  $\gamma$  is set to 1 by default. If the receptive fields of  $w_i$  and  $w_j$  are uniformly distributed (they do not favor any cluster), their connectivity strength is  $|RF_i|/n_i + |RF_j|/n_j$  by definition. If  $w_i$  and  $w_j$  are in the same cluster, their connection to one another is stronger than to other neighbours, which corresponds to  $\gamma=1$ .  $\gamma>1$  could increase granularity since it raises the merge threshold. Neighbours (and accordingly,  $n_i$  and  $n_j$ ) could alternatively be defined based on connectedness in the Delunay triangulation instead of SOM locations. This, however, may yield too many neighbours for noisy data and may necessitate a threshold for “important” neighbours. We will explore this in future work. After this phase, decision can be made about the interpolating units that were excluded before tree construction. We regard them as singleton clusters and perform merge according to Phase IV, Case IV/2.

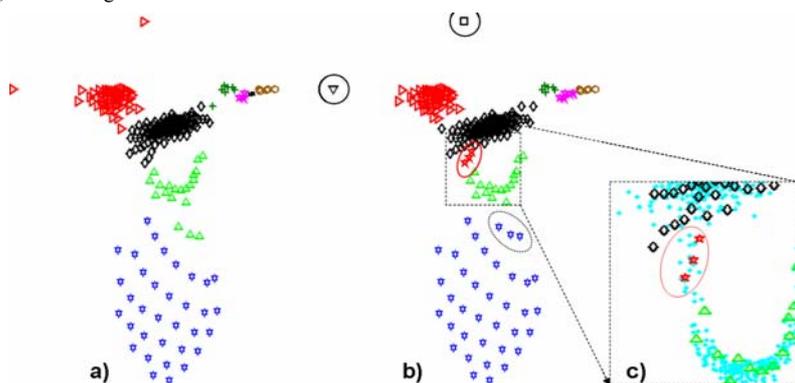
### 3 Experiments and Results

#### Clown Data

We chose this 2-d data set, used in [13], for performance comparison with [13]. It consists of 2220 samples grouped into several cluster types such as spherical (left eye), elliptical (nose), U-shaped (mouth), sparse (body), a cluster with three subclusters (right eye), and outliers (Figure 2, Left). Although the KSOM in [13] clearly separates the encircled outlier weights by empty PEs (Figure 2, Right), their method does not identify them because [13] does not allow singletons (Figure 3. a). Our use of single linkage distance and the connectivity matrix results in two separate singleton clusters for these outliers (Figure 3. b). The user can discard, or accept them as rare clusters. We discover a new cluster, shown by red stars in Figures 3. b, c), consistent with the slight gaps separating it from both the nose and the mouth. It is validated quantitatively by failing the criterion of eq. (4) for merge with either the nose or the mouth.



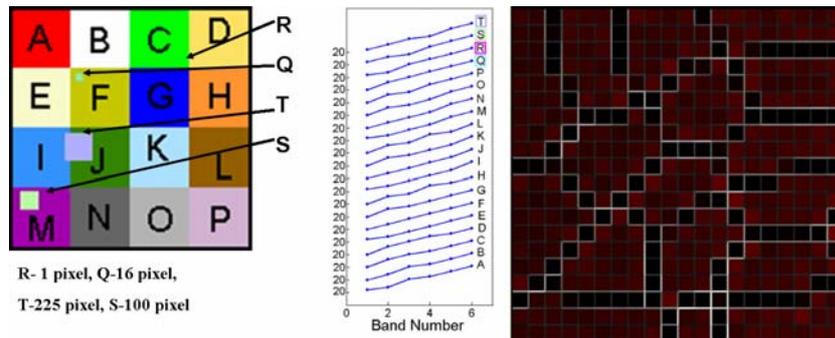
**Figure 2.** Left: 2-d Clown data with several cluster types : spherical (left eye), elliptical (nose), U-shaped (mouth), sparse (body), a cluster with three subclusters (right eye), and outliers. The inset shows finer cluster between nose and mouth. Right: KSOM weights overlain the data (cyan dots). Black dots and x symbols are weights with non-empty and empty receptive fields, respectively. Circled are outlier weights, clearly separated from other clusters by empty PEs. We thank E. Alhoniemi, at the University of Turku, for sharing the Clown data.



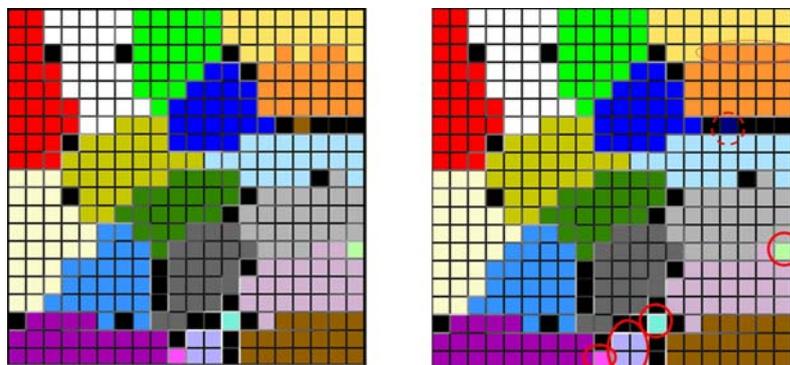
**Figure 3.** Clustering of the SOM weights of the Clown data. Each symbol is one weight and different symbols show different clusters. a) Clustering by [13] groups outliers into the closest cluster, i.e., groups the stray right pointing triangles with the left eye. b) Our clustering captures those as different clusters (circled weights). We correctly cluster the 3 weights (circled open blue stars) in the body that are clustered as mouth in a). We also find a new cluster, (circled red stars) between the nose (diamonds) and the mouth (upright triangles), shown in Figure 2. c) Blow-up of the nose and mouth area with data points underlying in cyan.

#### 20-class Synthetic Data

This 6-band 128 x 128 pixel image contains 20 classes 4 of which are rare, R has only 1 point. The classes were generated by adding 10% random Gaussian noise to mean signatures (Figure 4). Our method finds all classes correctly, with some minor exceptions explained in Figure 5.



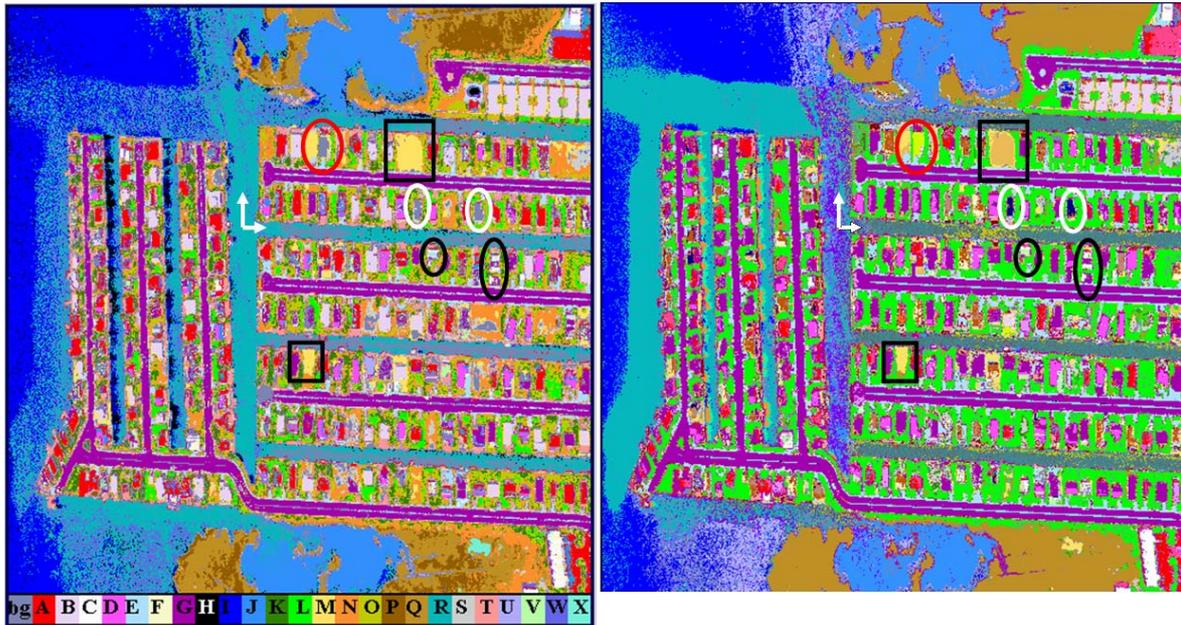
**Figure 4.** Left The 20 classes in the synthetic image. Middle: The mean signatures of the classes, offset for clarity. The subtle spectral differences and rare classes pose a clustering challenge. Right: U-matrix of the 20 x 20 SOM. Hit counts are indicated by the intensity of the red colour of the cells and are apparently low for interpolating weights in the inter-cluster gaps.



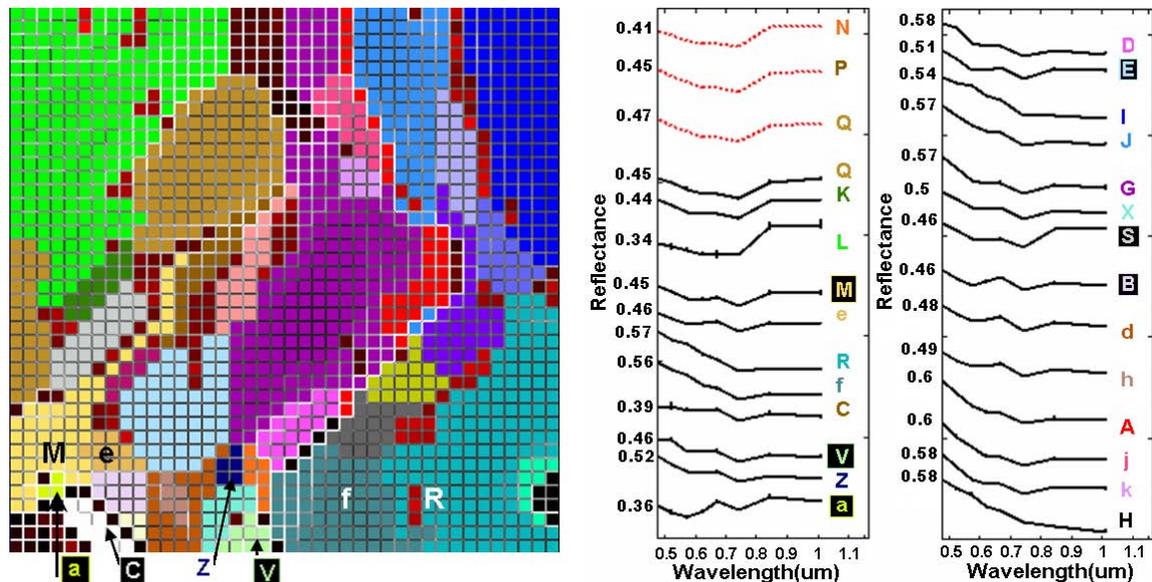
**Figure 5.** Left: The known class labels superimposed on the SOM. Black cells are empty PEs. Right: Our clustering correctly identifies all classes, including the rare ones (in solid red circles). The one-pixel class R is pink. The dark blue cell in the dashed red circle is a noisy outlier (does not belong to any class although its original label is L (brown)). Some weights with original D labels (yellow) are clustered as H (orange, in dotted oval) but close inspection reveals that their noisy signatures are closer to H.

**Ocean City Data**

An urban remote sensing spectral image of Ocean City, Maryland, serves for the verification of our method on real data. This 512 x 512 image has 8 spectral bands (each pixel is represented by an 8-d features vector spectrum). Detailed identification of clusters and discovery of rare clusters in such images are of great interest. An earlier supervised classification (Figure 6. Left) involving 24 classes, and a previous semi-manual clustering of the same SOM which found a new rare class in addition to those in the supervised map (see [16] for details), provide performance comparison. Figure 6 highlights the main points. Our cluster map has a general agreement with the supervised class map. Our method finds the previously identified rare clusters (greenish-yellow and white in Figure 6, Right, and in Figure 7), and also discovers a new rare class (in the white ovals in Figure 6, Right, and in Figure 7) which is unclassified in the supervised map (Figure 6, Left), and is not extracted by semi-manual clustering [16]. Its spectral signature is distinct from the rest. Figure 7 shows subcluster structures that refine clusters (M (yellow) and R (turquoise) in both the supervised and semi-manual cluster map in [16]. The signatures of the subclusters are similar to that of the parent cluster with slight but consistent differences. One problem in our clustering is exemplified by the merging of classes Q (brown), P (dark brown) and N (orange) into one supercluster (Q, two large brown areas in Figure 6, Right). Class P and Q are not distinguished in the semi-manual clustering either, but Q and N are extracted there as two separate clusters. Examination of the weights shows that although the mean signatures of classes Q and N are different, the weights exhibit a smooth changeover from one type to the other. This causes the undesirable merge by our algorithm.



**Figure 6. Left:** An earlier supervised classification of the Ocean City image, mapping 24 known cover types. Red and white ovals show unclassified shapes of buildings (the colour of the background, 'bg'). The black ovals show one rare class that was included in the supervised classification. **Right:** Clusters identified by our method. The agreement between the cluster map and supervised class map is very good. While the supervised class map has many unclassified pixels, here the vast majority of the pixels are assigned to clusters, which produces more appearances of some colours such as green and turquoise. The unclassified gray spots (in red and white ovals on the left) are now filled exactly, and with colours (greenish-yellow, label a in the red oval, and dark blue, Z in the white ovals) different from the existing 24 colours. See Figure 7 for labels. Their spectral signatures are distinct from the rest (Figure 7). The new-found rare clusters only occur at these locations. We also found subclusters. One example is in the black rectangles (the supervised class M is split into M and e). Arrows indicate two subclusters (turquoise R and dark turquoise f) of the former turquoise class. R coincides with the water in the canals open to the ocean while the water cluster f tends to be in canals farther in the city. See Figure 7 for the clusters in the SOM, and for their mean signatures. We thank Dr. Bea Csathó, Ohio State University, Byrd Polar Institute, for the Ocean City data and ground truth.



**Figure 7.** Clustering of the 40 x 40 SOM of Ocean City. **Left:** Labels of 32 clusters produced by our method overlain the U-matrix. Arrows show the rare clusters. Parent cluster M in Figure 6, Left, and in earlier clustering [15] splits here into M and e. Similarly, former cluster R splits into R and f. **Right:** Mean signatures of selected clusters, offset for clarity. The standard deviations (vertical bars on the signatures) are very small, an indication of the clustering quality. The signatures of the rare clusters (C, V, Z, a) are distinct. Subclusters M, e, R, f show slight but consistent spectral differences. Dotted lines at the top of the left panel are signatures of earlier supervised classes P, Q, and N, which are grouped into one super-cluster, Q, in our clustering.

## 4 Conclusions

We presented a promising algorithm for the clustering of trained SOMs, which we believe is an improvement over existing similar methods. The strength of it is in the exploitation of topology, in addition to the use of local data density and the choice of distance and cluster validity metrics. Our preliminary results demonstrate successful structure extraction from several data sets, including a large real multi-band image, all with clusters of varying properties, and including the discovery of rare clusters and subclusters. Our algorithm is highly automated requiring only 2 user-set parameters. A weakness is unduly merging of some larger clusters that show a smooth changeover from one pattern type to the other. Future work will address this.

## References

1. T. Kohonen (1997), *Self Organizing Maps*, Springer, Second Edition
2. A. Ultsch (1993), Self-organizing neural networks for visualization and classification, *Information and Classification — Concepts, Methods and Applications*, (O. Opitz, B. Lausen, R. Klar, Eds.), Springer Verlag, Berlin, p. 307-313.
3. M. A. Kraaijveld, J. Mao and A. K. Jain (1995), A nonlinear projection method based on Kohonen's topology preserving maps, *IEEE Trans. Neural Networks*, **vol. 6**, p. 548-559.
4. D. Merkl and A. Rauber (1997), Alternative Ways for Cluster Visualization in Self-Organizing Maps, *Proc. Workshop on Self-Organizing Maps (WSOM'97)*, Espoo, Finland, p. 106-111.
5. A. Ultsch (2003), Maps for the Visualization of High-dimensional Data Spaces, *Proc. Workshop on Self-Organizing Maps (WSOM'03)*, p. 225-230.
6. M. Cottrell, E. DeBodt (1996), A Kohonen Map Representation to Avoid Misleading Interpretations, *Proc. European Symposium on Artificial Neural Networks*, Bruges, Belgium, D-Facto, p. 103-110.
7. J. Himberg (2000), A SOM based cluster visualization and its application for false colouring, *Proc. IEEE-INNS-ENNS Int'l Joint Conf. On Neural Networks*, Como, Italy, **vol. 3**, p. 587-592.
8. S. Kaski, T. Kohonen, J. Venna (1998), Tips for SOM Processing and Colourcoding of Maps, *Visual Explorations in Finance Using Self-Organizing Maps*, (G. Deboeck, T. Kohonen, Eds.), London
9. T. Villmann, E. Merényi (2001), Extensions and Modifications of the Kohonen SOM and Applications in Remote Sensing Image Analysis, *Self-Organizing Maps: Recent Advances and Applications* (U. Seiffert and L.C. Jain Eds.), Springer-Verlag, p. 121-145.
10. J. Blackmore and R. Miikkulainen (1995), Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network, *Machine Learning: Proceedings of the 12th International Conference* (Tahoe City, CA), (A. Prieditis, S. Russell Eds.), San Francisco: Kaufmann, p. 55-63.
11. H. Lipson and H.T. Siegelmann (2000), Clustering Irregular Shapes Using High-Order Neurons, *Neural Computation*, **vol. 12**, no. 10, p. 2331-2353.
12. M. Cottrell, P. Rousset (1997), The Kohonen Algorithm: A Powerful Tool for Analyzing and Representing Multidimensional Quantitative and Qualitative Data. *IWANN 1997 (International Work-Conference on Artificial Neural Networks)*, p. 861-871
13. J. Vesanto and E. Alhoniemi (2000), Clustering of the Self-Organizing Maps, *IEEE Trans. on Neural Networks*, **vol 11**, no. 3, p. 586-600.
14. F. Murthag (1995), Interpreting the Kohonen self-organizing map using contiguity-constrained clustering, *Pattern Recognition Letters*, **vol 16**, p. 399-408.
15. A. Jain (2004), Issues Related to Data Mining with Self-Organizing Maps, *M.Sc. Thesis*, Electrical and Computer Engineering, Rice University, Houston, Texas.
16. E. Merényi, A. Jain (2004), Forbidden Magnification? II, *Proc. 12<sup>th</sup> European Symposium on Artificial Neural Networks (ESANN'04)*, Bruges, Belgium, D-Facto, p. 57-62.