Matthew Kyan, Kambiz Jarrah, Paisarn Muneesawang, and Ling Guan
Ryerson University, CANADA

Strategies for Unsupervised Multimedia Processing

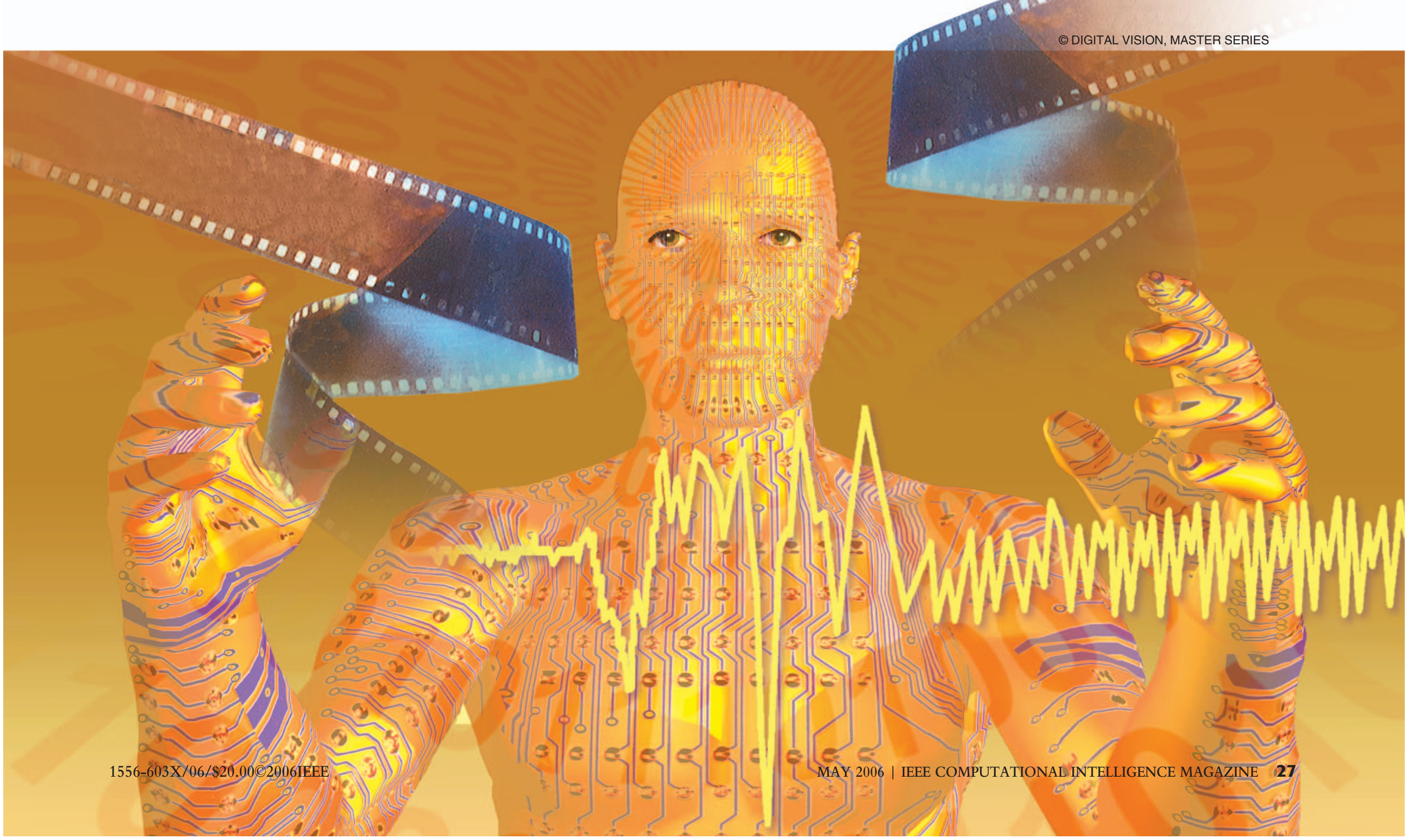
Self-Organizing Trees and Forests

As the fruit of the Multimedia Age comes to bare, the question of how such information might be effectively harvested, archived and analyzed, remains a monumental challenge facing today's research community. The processing of such information, however, is often fraught with the need for conceptual interpretation; a relatively simple task for humans, yet arduous for computers. In attempting to handle oppressive volumes of information becoming readily accessible within consumer and industrial sectors, some level of automation remains a highly desired goal.

To achieve such a goal requires computational systems that exhibit some degree of intelligence in terms of being able to formulate their own models of the data in question with little or no user intervention. Such systems must be able to maintain basic decisions about what information is actually important and what is not. In effect, like the human, the system must be able to *discover* characteristic properties from within the data in some appropriate manner. This process is known as Unsupervised Learning—sometimes referred to as *Clustering* or *Unsupervised Pattern Classification*—an essentially pure form of *Data Mining*.

Computational technologies based on Artificial Neural Networks have been the focus for much research into the problem of unsupervised learning; in particular, network architectures that are based on principles of Self-Organization. Such principles are, in many ways, centered on Turing's initial observation in 1952 [1], namely that *global order can arise from local interactions*. Receiving much neurobiological support, such mechanisms are believed to be fundamental to the organization that takes place in the human brain.

In this article, we explore a new family of neural network architectures that have a basis in self organization, yet are



© DIGITAL VISION, MASTER SERIES

1556-603X/06/$20.00©2006IEEE

somewhat free from many of the constraints typical of other well-known self-organizing architectures. Within this family, the basic processing unit is known as the Self-Organizing Tree Map (SOTM). We will look at how this model has evolved since its inception in 1995, how it has inspired new models, and how it is being applied to complex multimedia research problems in Digital Asset Management & Microbiological Image Analysis.

I. Unsupervised Learning and Self Organization

The goal of Unsupervised Learning is to essentially formulate or discover significant patterns or features in a given set of data, *without* the guidance of a teacher. The patterns are usually stored as a set of prototypes or clusters: representations or groupings of *similar* data.

In describing an unknown set of data, such techniques find much application across a wide range of industries, particularly in bioinformatics (clustering of genetic data, protein structure analysis), image processing (segmentation, image retrieval), and other applications that warrant a significant need for data mining.

Unsupervised Learning & Self-Organization are inherently related. In general, self-organizing systems are typified by the union of *local interactions and competition over some limited resource*. In his book [2], Haykin identifies four major principles of self-organization (see Figure 1):

❏ Synaptic Self-Amplification
❏ Synaptic Competition
❏ Co-operation
❏ Knowledge through Redundancy

A. Synaptic Self-Amplification and Competition

The first principle of self organization is expressed through Hebb's postulate of learning [3], which is supported by neurobiological evidence. This states that (on the cellular level) when two cells are within significant proximity enabling one to excite another, and furthermore, do so persistently, some form of physiological/metabolic growth process results. This process works to enhance the firing cell's efficiency in triggering the second cell. In other words (in neural network terms), a synaptic path evolves and strengthens between the two neurons (memory cells), so that future associations occur much more readily. This action of strengthening the association between two nodes functions as a correlation between their two states.

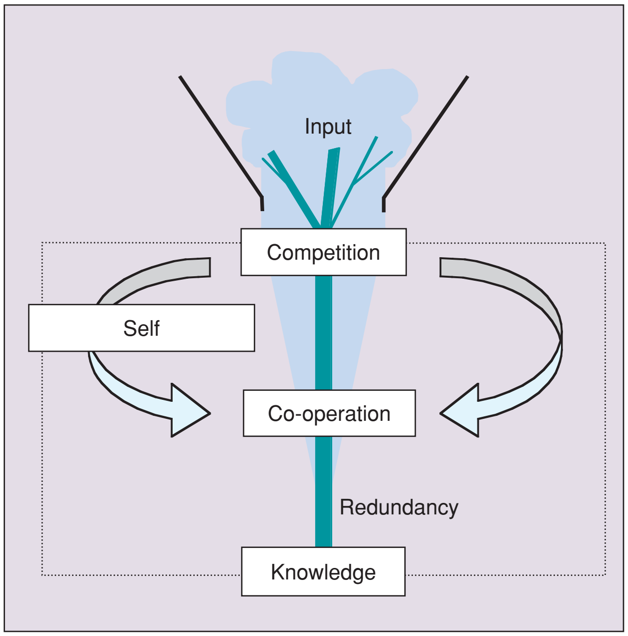


FIGURE 1 The four principles of self-organization—work to filter and encode the redundant (ordered/structured patterns from a set of input data).

Typically, the Hebbian learning rule that ensues from this principle would eventually drive synapses into saturation were it not for the second principle, wherein some *competition* occurs for limited resources. These properties combined, have led to the modified Hebbian *adaptive* rule, as proposed by Kohonen [4]:

$$\Delta w_{k^*} = \alpha\varphi(k^*, x_i)[x_i - w_{k*}], \tag{1}$$

where w_{k^*} is the synaptic weight vector of the *winning* neuron k^* (see below), α is the learning rate, and **some scalar response $\varphi(\cdot)$ to the firing of neuron k* (activation)**.

The activation function $\varphi(\cdot)$ is the result of the second principle (competition). Neurons generally compete to see which is most representative of a given input pattern presented to the network. Some form of discriminative function oversees this process (for example, choosing a neuron as a *winner* if its synaptic vector minimizes Euclidean distance over the set of all neurons). This process is often termed *Competitive Learning*.

B. Co-Operation

Hebb's postulate is also suggestive of the lateral or associative aspect to the way knowledge is then captured in the network, i.e., not just through the *winner*, but also through nearby neurons in the output layer. This property is generally implemented in self-organizing architectures by virtue of the way in which nodes are interconnected. Often the strength of connection is not considered, but rather a simple link functions as an indicator of which other nodes in the network will more readily be associated with a winning node. Thus a local neighborhood is defined, and it is through this that knowledge may be imparted. Local adaptation usually follows the simple Kohonen update rule, whereby a portion of information is learned by the winning node, with neighboring nodes extracting lesser portions from the same input.

C. Knowledge Through Redundancy

Although not explicitly encoded, the final principle of self-organization implicitly results from the action of the first three. When exposed, any order or structure inherent within a series of activation patterns represents redundant information that is ultimately encoded by the network as knowledge. In other words, the network will evolve such that similar patterns will be captured and encoded by similar output nodes, while neighboring nodes organize themselves according to these dominant redundancies, each in turn focusing on and encoding lesser redundant patterns across the input space.

D. The Kohonen Self-Organizing Map (SOM)

Architectures such as Kohonen's *Self-Organizing Map* (SOM) [4] represent one of the most fundamental realizations of these principles, and as such have been the foundation for much research in data mining.

Associative connections linking prototypes within SOM-based clustering algorithms are generally responsible for their innate ability to infer an *ordered* or *topologically-preserved* mapping of the underlying data space. Associations among nodes are advantageous as they help guide the evolution of such networks, and may also assist in formulating post-processing strategies or for extracting higher-level properties of any clusters discovered (e.g., inter-cluster relationships). This type of property is often used for the visualization of high-dimensional data—where multivariate data (of dimension greater than 3) is mapped onto a two-dimensional grid such as the lattice of a SOM. Since topology is preserved, neighboring nodes in the lattice are often representative of related properties in the original data [5].

In typical SOM-based approaches, topology is imposed rather than inferred (via a fixed predefined lattice of nodes). Unsupervised learning, however, is ill-posed: the nature of the underlying data is unknown, thus it is difficult to infer what an appropriate number of classes might be, or how they should be related.

Dynamically generating self-organizing networks attempt to address this issue by formulating a set of dynamic associations as they grow to represent the data space.

II. Dynamic Architectures for Self-Organization

Among the many proposed dynamic extensions to the classic SOM algorithm, there exist two principal approaches: *hierarchical* and *non-stationary*, as well as *hybrids* of these two. Many of these structures have foundations in various stationary methods including the SOM itself (see Figure 2), *Competitive Learning* (CL) [6], [7], *Neural Gas* (NG) [8], or the *Hierarchical Feature Map* (HFM) [9]. Although they form a basis for many of the dynamic architectures highlighted in this section (see Figure 3), stationary models of themselves are limited as SOM is, in terms of the number of possible classes being constrained at run-time. Nevertheless, they serve as important precursors to the following.

A. Hierarchical Architectures

This class of approaches attempts to foster the extraction of hierarchical relationships: groupings based on differing levels of granularity within the data to be clustered.

Such models are based on principles similar to the HFM, wherein a pyramidal hierarchy of SOM's are trained in a top-down manner. Each unit in the SOM of an upper layer is connected to its own SOM map in a lower layer, where the data space for training lower level maps results from individual portions of the original data space, as partitioned by higher level SOM's. An appropriate choice of network parameters that can handle different datasets remains elusive, thus any classification is inherently bound to the network structure rather than the underlying data distribution.

In the *Growing Hierarchical SOM* (GHSOM) [10]–[12], an initial SOM is grown until a map-quality criterion is satisfied, after which SOM's are branched according to quality measures achieved at higher levels, where the process is continued. Growth thus proceeds in a layered fashion.

Efficient computation is achieved when utilizing relatively smaller maps in upper levels of the hierarchy (where training data is large), and larger maps in lower levels of the hierarchy. This allows for finer resolution of detail in the more confined regions of the original data space. Such approaches have been used for such tasks as the mining and archiving of text documents, where small maps in higher levels of the hierarchy form reference to course grain topics, while lower level maps more readily specialize [13].

The *Tree Structured SOM* [14] offers a computational boost by building a hierarchy for efficient fast-winner search. Similar principles are adopted in the *Evolving Tree* algorithm [15], wherein the frequency with which a given node is fired accumulates, acting as an indicator for subdivision of the network. Subdivision is thus controlled by a threshold, which decays during time to hinder the growth of the tree according to a fixed regularization parameter. New nodes (a predefined number) then form children of the original winner, which is subsequently retired from competition, retaining its memory for use in a top-down search for future winning nodes.

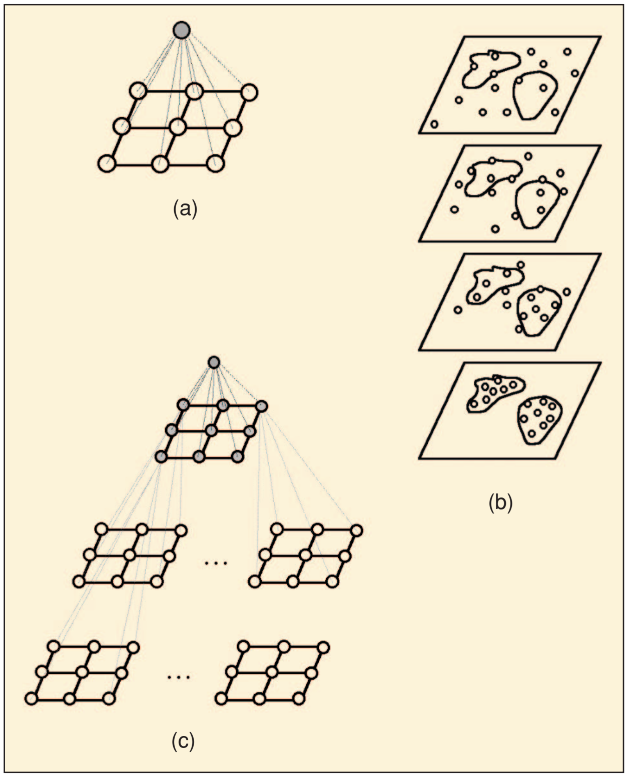


FIGURE 2 Classic stationary competitive learning architectures: (a) Kohonen's SOM; (b) neural gas; (c) hierarchical feature map.

One of the problems with these approaches, however, is that nodes from differing branches (lower in the hierarchy), may in fact move closer to one another during later iterations. In doing so, higher resolution clusters that are initially targeted may lead to errors in matching winners with the current input. One option is to subdivide the input space along with the tree network. This would resolve the search dilemma; however, errors in course segmentations at higher levels of the hierarchy may result in parts of natural clusters in the data at higher resolutions being disassociated from one another within the hierarchy. Clearly such approaches need a mechanism to redistribute data (and/or nodes) among sub-trees within the overall hierarchy as needed.

B. Non-Stationary Architectures

The well-known *Neural Gas* mechanism has spawned a family of dynamic network variants: *Growing Cell Structures* (GCS) [16], *Growing Grid* (GG) [17] and *Growing Neural Gas* (GNG) [18]. Each of these algorithms follow a subdivision strategy based upon the consideration of accumulated error measures and are evaluated at regular intervals during network evolution.

Topological information is simultaneously captured through a *Competitive Hebbian Learning* (CHL) mechanism. In this process, the Hebbian principle of associative memory has been taken a step further than described in Section I.B. In [19], strength of association is also utilized for the dynamic adaptation of neighborhoods themselves. This process uses an edge formation, ageing and removal scheme to emulate the strengthening and decay of node associations, resulting in the discovery of the underlying topology of data. This topology can then be exploited along with accumulated error information for the selective insertion of new nodes into the network. Existing nodes possessing the highest accumulation of error at the time of evaluation are split until a maximum error criterion is satisfied by all nodes within the network. GNG also may implement a strategy to drop nodes formed, should they ever become completely disassociated (having no connection with any other node).

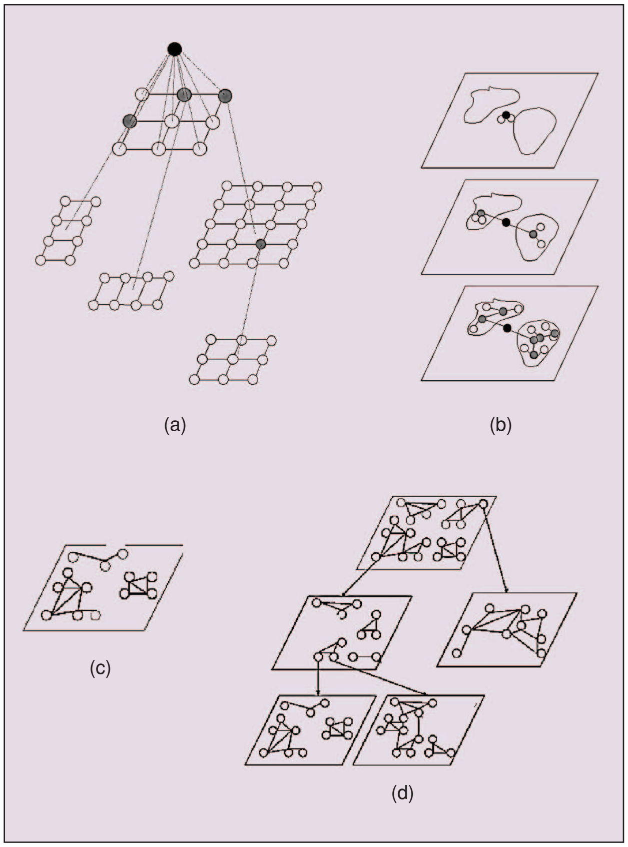


FIGURE 3 Dynamic competitive learning architectures: (a) GHSOM; (b) Evolving Tree; (c) GNG; (d) DASH.

Due to the online nature of measuring error statistics, these methods simulate 'average' distortion error within each node, by accumulating then redistributing error between the original error prone nodes and their inserted 'offspring.' Error values throughout the network are 'relaxed,' or decayed during time to emphasize the accumulation of more recent errors as opposed to those accumulated due to past network states.

An alternative mechanism for deducing 'average' distortion errors might be to incorporate the *Conscience Learning* mechanism [20] as a relative measure of probability density (more akin to an online measure of frequency with which a neuron wins competition in the network). In addition, conscience learning may yield a better estimate of candidates for node removal than edge information alone (particularly in GNG variants).

Error-based mechanisms for partitioning the input data space seem more intuitive than those based on density alone, as some consideration of the *similarity* among data points associated with a given node directly impacts the formation of new cluster prototypes. By inserting nodes to reduce such error, the network naturally works to reduce the mean square error among cluster prototypes and the original data. This process can be allowed to continue unchecked of course, causing unlimited growth of the network—limited in size only by the original data space itself. Such growth is obviously undesired, thus the relaxation of error statistics serves a secondary purpose—that of hindering the growth of the network.

When considering error alone as a trigger for node insertion, there is the implication that desired clusters are of a similar size (i.e., degree of similarity). To an extent, similar density is also implied, as error is evaluated as an accumulation. In other words, it is anticipated that different groupings of data within some feature space, group to the same *degree* of similarity. For many applications this may be a valid assumption to make, for others it may not (e.g., locating two spatial Gaussian distributions of differing standard deviation).

Error-driven processes may also lead to the premature assignment of *dead nodes* (nodes that become trapped in low density regions such as outliers or noise), as a blind insertion mechanism attempts to force a node between two existing nodes with significant error. This problem is alleviated somewhat by restricting the insertion such that it only occurs between *connected* nodes of high error, under the assumption that connections only form when some underlying density has been detected in the region between the nodes.

It is anticipated that nodes inserted in this manner will 'track' back to regions of higher density in due course, as a result of Kohonen learning. The ability of a node to be pulled out of local minima, however, becomes more difficult as the network grows and more nodes compete for dense regions of the data space. In GNG and its variants, the *distortion* resulting from the proliferation of dead nodes is very sensitive to the regularity (some interval of iterations: λ) with which the network is evaluated and node insertion enforced.

Error-driven methods also do not necessarily work to capture global topology at all levels of network evolution but rather unfold or unravel through the data space to approach a more global description of the topology. Thus, little to no hierarchical information eventuates.

One hybrid mechanism that has been proposed that attempts to blend hierarchical information of GHSOM with dynamic growth properties of GNG is the *Dynamic Adaptive Self-Organizing Hybrid* (DASH) model [21]. With non-stationary properties of GNG, this model features some level of tolerance to maintaining hierarchical relationships, yet is able to adjust to new information while forgetting old information that has become irrelevant. In this model, layers of GNG's are essentially formed; however, within each, traditional limitations of the GNG are still evident.

In the remainder of this article, the Self Organizing Tree Map (SOTM) [22] and models arising thereof is explored. In principle, SOTM would typically be grouped in the non-stationary family; however, it grows and encodes hierarchical information naturally, thus can be considered as belonging to both.

III. The Self-Organizing Tree Map (SOTM)

The SOTM was first proposed in 1995 [22] and later developed [23], [24]. In the basic SOTM process, *Competitive Learning* is implemented just as in the SOM; however, the structure of the network is dynamic and is grown from a single node. In many ways, the SOTM process draws inspiration from Adaptive Resonance Theory (ART) applied to an SOM-type framework.

In ART [25]–[27], the network has an outer layer that is capable of storing a certain capacity of prototypes (output layer). With each input, competitive learning locates the nearest prototype in the outer layer, and a test is performed to establish how similar the input is to the existing prototype(s). This test is known as a vigilance test. If the input is within a tolerance of the winning prototype, then *resonance* is said to occur, resulting in the refinement of the winning prototype. Otherwise, while there is still capacity, a new prototype is formed.

Node or prototype insertion in the SOTM is driven by a top-down process that effectively explores the data space from the *outside-in*. An ellipsoid of *significant similarity* [28] forms a global vigilance threshold $H(t)$ that is used to evaluate the proximity of each new input sample against the closest existing prototype (winning node) currently in the network. A hard decision is made: if the input is distant from the winner beyond the threshold, a new prototype is spawned as a child node to the current winner, and the process continues with the next input. Otherwise, the input is deemed *significantly similar* and the winner is updated toward the input. This insertion process is shown in Figure 16(a) (see [28] for a more detailed treatment).

It is through this dynamic vigilance function, Kohonen style adaptation and topological connections, that the SOTM may be distinguished from ART-based models. In fact, this vigilance can be thought of as a form of *hierarchical control* [22], as it evaluates proximity at decreasing levels during time. In effect, this works to form a hierarchical partitioning of the data. The hierarchical approach naturally attempts to maximize discrimination across the entire data space in early phases of *discovery* before considering finer differences in the data.

Error-driven mechanisms such as GNG, GCS and GG do not necessarily achieve this, as they unfold outward—expanding across the data space. As such, if the network was to exhibit a limited capacity for prototypes (i.e., be limited in the number of classes it uses), it is quite possible that nodes generated may not form a representation that adequately spans the data.

Due to the relationship among new nodes and their parents (winners at the time of generation), a tree structure naturally evolves during SOTM growth. This may also operate as an indicator of neighboring nodes, as they are generated according to proximity, as differing levels of granularity are parsed. This can be seen in Figures 4(a)-(b), which also show a comparison of the clustering efficiencies between the SOTM and the SOM on a two-dimensional feature space [23]. In these figures, the input vectors are uniformly distributed within many rectangular squares. The SOTM's clustering performance in Figure 4(a) shows that there are no neurons lying in a zero-density area. In contrast, although the SOM's topology exhibits the distribution of the structured input vectors, it also introduces several false

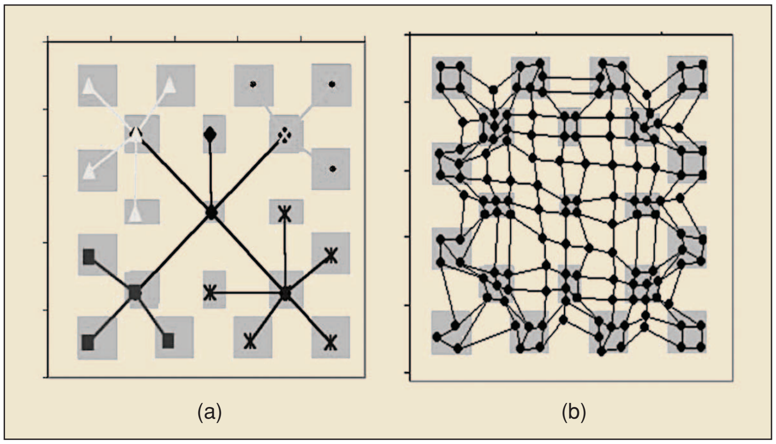


FIGURE 4 Self-organizing data clustering: (a) performed by SOTM, no nodes converge to join the areas of zero data density; (b) performed by SOM, nodes converge to areas of zero data density.

representations outside of the distribution of the input space, as shown in Figure 4(b).

As indicated by this example, the SOTM realizes the learning such that its connections organize to form a suitable, efficient network structure. This property is necessary for a high-dimensional input space with a sparse data structure, where it is important to prioritize the spanning of data in the current application, so as to classify image relevancy.

IV. Application of SOTM to Image Retrieval

The goal of image retrieval is to find, from a large image database, a set of images whose contents are similar to a given query. Each image in the database is considered as a vector point in the feature space that characterizes its color, shape, and texture information. A similarity function such as Euclidean distance is applied to measure similarity among vectors.

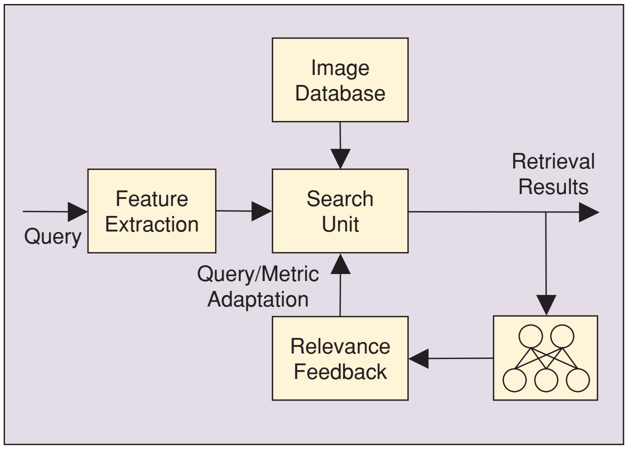


FIGURE 5 Machine-controlled relevance feedback system using SOTM.

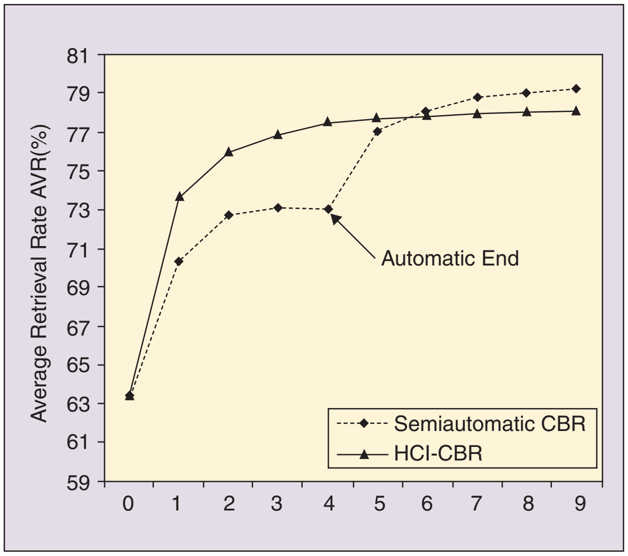


FIGURE 6 A comparison of retrieval performance at convergence, between the semiautomatic and HCI-CBR (human-controlled interactive content-based retrieval) methods. The database used is Brodatz texture database. The results were averaged over 116 queries.

In order to reduce the gap between low-level features used for image indexing and the high-level concepts used by users, relevance feedback-based machine learning techniques were introduced [29], [33]. In these techniques, a user provides initial query and trains (supervises) a search engine on what is regarded as relevant images through relevance feedback learning. The search engine is considered as a supervised learning unit that adapts itself according to user feedback.

There are two problems, however, associated with this learning approach. First, the system requires a high degree of user workload in providing feedback samples through many cycles of relevance feedback before convergence. Second, the possible human subjective error is introduced when users provide relevance judgment to retrieved images.

In view of the above problems, SOTM was applied to minimize user workload by implementing automatic relevance feedback process [30], [31]. User interaction is replaced by SOTM (see Figure 5), which guides the adaptation of relevance feedback module.

This implementation can run in automatic mode as well as semiautomatic mode, thereby allowing the system to capture different levels of subjectivity among users. Figure 6 shows a comparison of performance between human-controlled relevance feedback and SOTM-based automatic relevance feedback [31].

Two important observations are: the retrieval performance is significantly improved as a function of feedback iterations; and the semiautomatic learning method reduces the amount of user feedback required for convergence as compared to human-controlled relevant feedback method.

One major advantage of the SOTM over classical unsupervised learning methods, such as SOM and K-mean algorithms, is its ability to work well with sparsely distributed data [23]. SOTM is chosen in the current application, as the problems in image retrieval have different characteristics than other applications of data classification. First, the training data set, required by relevance feedback learning algorithms, is very small, e.g., a few to ten samples. Also, the feature space is of a very high dimension consisting of a combination of color, shape, and texture features. These tend to form sparsely distributed data.

Second, a problem is caused by an unbalanced data distribution between relevant and non-relevant samples in the training set. It is expected that, after the first iteration of relevance feedback, the relevant items are retrieved more than non-relevant items, and thus, the majority of relevant items will introduce an unbalanced space to the resulting clusters. To solve this problem, the SOTM allows for a focus and maximization of discrimination within sub-regions of the (unbalanced) training data via competition among its hierarchically-discovered nodes. This efficient allocation and breakdown of class relationships minimizes classification errors as compared to that achieved through SOM, GNG or similar, which unfold across data space, often leading to distortion within sparse interstices.

Early implementations of SOTM for automatic relevance feedback were focused on compressed-domain processing [30], [31]. Due to a high volume of compressed data in databases, it was proposed to extract image descriptions directly from compressed images (via wavelet or DCT transformations) to avoid complexity of decompression.

These descriptors are employed for fast processing in retrieval. The time-consuming decompression step is only applied to the retrieved images (i.e., training samples). Then a feature extraction process in the spatial domain is applied to obtain image descriptors of higher quality for relevant classification via SOTM.

Incorporating knowledge to decide image relevancy is indeed important to increase retrieval performance in automatic relevance feedback mechanism. This suggests that the quality of low-level descriptions used in describing image content need some injection of knowledge to achieve a better weighting scheme for discrimination in SOTM-based relevance classification [32]. Region of interest (ROI) is a form of basic knowledge in the domain of photograph collection. An ROI is assumed to be an area around the center focus of a camera in photography. The features extracted within the ROI provide better discrimination ability to the self-organizing network.

V. The DSOTM for Enhanced Automation in Content-Based Image Retrieval

The main focus of this section is to provide an enhanced methodology for reducing the gap between low-level descriptors and high-level concepts (i.e., semantics), such that a CBIR system can achieve further improved performance while in automatic retrieval mode. A genetic algorithm (GA) based approach is used in conjunction with a recent addition to the SOTM family, the Directed Self-Organizing Tree Map (DSOTM). The resulting framework is referred to as GA-based CBIR (GA-CBIR), and aims to generalize image retrieval application to real-life photos stored in more generic digital archives. GA-CBIR mimics the human-feedback mechanism proposed in the previous section, by automatically adapting the search process according to what the system evolves to believe is significant content within the query. In this engine, DSOTM is a major component of the GA analysis as it closely works with traditional GA operators to achieve a more robust perceptual classification—one that is more in tune with the apparent discriminative characteristics observed in an image by a human user.

A. The Directed Self-Organizing Tree Map (DSOTM)

The SOTM algorithm not only extracts global intuition from an input pattern space but also injects some degree of localization into the discriminative process such that maximal discrimination becomes a priority at any given resolution (or number of classes). Also, the ability of SOTM to span and force division in the extremes of the data in the early stages, delaying division of most similar aspects until later stages of learning, makes it essentially sensitive to the most dominant differences in the data and, thus, less prone to classification errors and more attractive to the retrieval applications. Even though the current algorithm of SOTM is proven to be effective in minimizing the user workload, it suffers two major problems: it unsuitably decides on the relevant number of classes in a relevancy identification task; and often loses track of the true query position.

In attempts described in the previous section, the decision on which clusters are relevant in the SOTM is postponed until after the algorithm has converged. This is because there is no innate controlling process available for the algorithm to influence cluster generation around the query center (the SOTM clusters entirely independent). Losing a sense of query location within the input space can have an undesired effect on the true structure of the relevant class and can force the SOTM algorithm to spawn new clusters and form unnecessary boundaries within the query class as is illustrated in Figure 7. Therefore, retaining some degree of supervision to prevent unnecessary boundaries from forming around the query class seems to be vital.

Due to above limitations, a new member of SOTM family, the Directed SOTM (DSOTM) has been proposed. The DSOTM algorithm not only provides a partial supervision on cluster generation by forcing divisions away from the query class but also makes a gradual decision on resemblance of the input patterns by constantly modifying each sample's membership during the learning phase of the algorithm. As a result, a more robust topology with respect to the query, as well as a better sense of likeness, can be achieved (Figure 8) [34].

On the other hand, DSOTM relies on the query position as the a-priori center of the relevant class and updates memberships according to this knowledge. Having said that, the synaptic vector adjustments in the DSOTM are not just limited to the winning node; the algorithm also constantly modifies all the centers according to the query position. Alternatively, if the winning center is not the query center, vector adjustments will affect both the winning node and

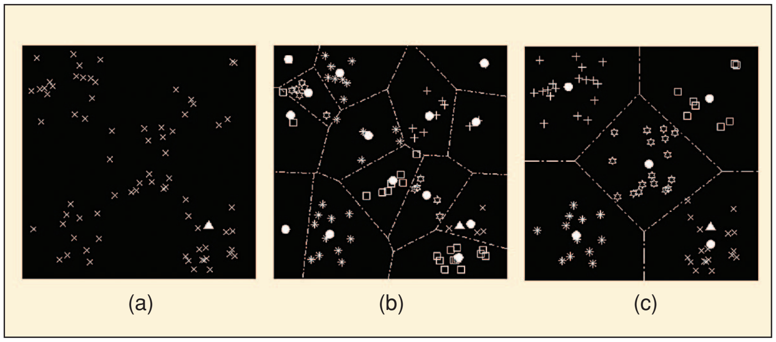


FIGURE 7 2-D mapping: (a) Input pattern with five distinct clusters, (b) 14 generated centers using SOTM, and (c) five generated centers using DSOTM. The SOTM forms a boundary near the query (triangle) contaminating relevant samples, where as some supervision is maintained in the DSOTM case, preventing unnecessary boundaries from forming.

the relevant center's position by moving the winning node (center of the irrelevant class) more toward the irrelevant samples and moving the relevant center more toward the query center. Thus, as more samples are exposed to the network, the DSOTM algorithm will learn less from irrelevant samples and more from the relevant ones by maintaining the integrity of a relevant center near the original query position. This action helps to foster a sense of *generalization* in the query class across feature subtleties that may objectively be deemed as discriminative by an independently operating SOTM. The DSOTM *desensitizes* partitioning in the region of a query.

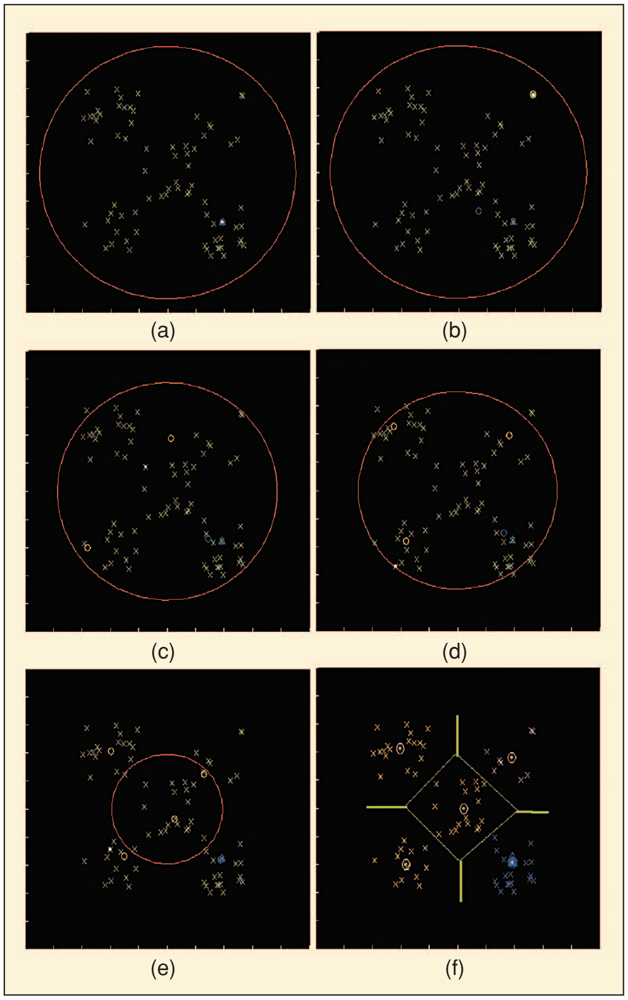


FIGURE 8 DSOTM classification in a two-dimensional feature space. (a)–(f) Each figure illustrates the current situation of the map, position of the query (triangle), state of the threshold (ellipsoid), and adaptation of the old centers as new nodes are generated. (f) Final state of the map.

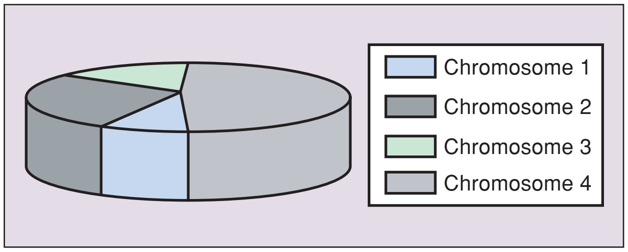


FIGURE 9 Probabilistic selection mechanism provided by GA roulette wheel.

B. Genetic Algorithm (GA)

The Genetic Algorithm is one of the most commonly used optimization techniques among a set of evolutionary computational algorithms and is inspired by Darwin's theory of evolution:

> *Individuals having advantageous variations are more likely to survive and reproduce than those without the advantageous variations.*

The formal idea of the algorithm was developed by John Holland in the late 1960s and early 1970s and continued by Goldberg in the late 1980s [35]. The basic concept of the algorithm is to model Darwin's evolutionary process where a population of *potential solutions* competes with each other for survival based on their inherent behavior. These potential solutions are usually encoded as strings called chromosomes, and each chromosome is associated with a fitness value that indicates its optimality with respect to the current optimization criterion. The fitness value of a solution will affect its probability of being selected for further evolution. Typically, chromosomes are binary strings of a fixed length, but other forms of strings are also used to encode the chromosomes. In this work, we use a base-10 representation of chromosomes. The goal of the evolutionary process is to generate new candidate solutions using the genetic operators: Selection, Crossover, and Mutation [36]–[39].

1. Selection

A *Roulette Wheel Selection* method is used to select chromosomes for reproduction in proportion to their score as indicated by a fitness function. *Roulette Wheel* is a probabilistic selection method, a technique in which a chromosome with higher fitness score in the population has a better chance to be selected for reproduction since it occupies a bigger slot on the Roulette Wheel (see Figure 9).

2. Reproduction

Crossover and Mutation are used to generate and manipulate a diverse number of individuals in the current population by allowing candidate solutions with higher fitness values a better chance to act as parents to generate offspring. In this way, they can pass desired traits of their own to the next generation. At the same time, they are able to evolve further themselves, to become even better solutions.

3. Crossover

Crossover is the main reproduction operator to generate new individuals in GA. The main idea of this operation is to combine segments of chromosomes of selected parents to form a new individual, which is similar to crossover process in genetics as is illustrated in Figure 10. Through this process, a new individual can inherit useful characteristics from its parents, which eventually leads to a higher fitness value.

4. Mutation

Changing the value of a gene in a randomly chosen position along a chromosome generated from Crossover is called mutation. Generally, mutation is regarded as a supplemental reproduction operator in GA, and the mutation probability is usually set to be very small. Therefore, the new selected individuals by mutation are not much different than their original parents (Figure 10).

TABLE 1 Experimental results in terms of *RR*.

QUERY SETS	CBIR WITH SOTM	CBIR WITH DSOTM	GA-CBIR WITH SOTM	GA-CBIR WITH DSOTM
A	47.5%	58.0%	66.8%	78.3%
B	47.4%	59.6%	72.1%	76.7%
C	51.0%	56.8%	74.4%	80.5%
Ave.	*48.6%*	*58.1%*	*71.1%*	*78.5%*

C. Genetic Algorithm & Automatic CBIR System (GA-CBIR)

The block diagram of the proposed GA-CBIR is illustrated in Figure 11. Since DSOTM seeks to provide a better judgment about resemblance of input samples to one another and since GA aims to select and reproduce a better fit population of candidate solutions, this DSOTM-GA combination can effortlessly replace the relevance feedback learning process to achieve a fully automated retrieval engine. The GA thus attempts to mimic human attention by working and coordinating the search *subjectively*, feeding and evaluating results from the more *objective* DSOTM. This is possible by highlighting dominant characteristics of images through the principle of feature weight detection. Such a detection mechanism seeks to emphasize certain characteristics in an image (i.e., its color, texture, and/or shape) that might provide significant information to the DSOTM for a more robust classification. Assigning proper weights for individual features can significantly increase performance behavior of CBIR systems, reducing the need for human supervision.

In this work, a number of experiments were conducted to compare behaviors of the automatic CBIR and GA-CBIR engines using SOTM and DSOTM clustering algorithms.

Simulations were carried out using a subset of the Corel image database consisting of nearly 5100 JPEG color images, covering a wide range of real-life photos from 51 different categories. Each category consisted of 100 visually-associated objects to simplify the measurements of the retrieval accuracy during the experiments. Three sets of 51 images were drawn from the database. In sets *A* and *B*, images were randomly selected from the entire set without regard for class; whereas in set *C*, images were randomly selected such that no two images were from the same class. Retrieval results were statistically calculated from each of the three sets. In the simulations, a total of 16 most relevant images were retrieved to evaluate the performance of the retrieval. The experimental results are illustrated in Table 1.

DSOTM outperforms the SOTM in both CBIR and GA-CBIR systems. Effectiveness of using genetic-based algorithm in the structure of CBIR systems is also evident in the following results. Figure 12 provides a vivid illustration of the GA-CBIR interface with the query image (house) on the top-left corner of each figure. The top figure represents the retrieval of the query using one iteration of DSOTM. All feature vectors are equally weighted to have a uniform contribution to the DSOTM classification.

Even though a retrieval rate of 62.5% for this particular query is achieved, this rate can be improved to 100% using the GA-CBIR engine (with respect to the capacity of retrieved results presented to the user). The bottom figure represents the retrieval of a similar query after the process of GA feature weight detection. As expected, the GA operation appears to identify the features important in aiding what humans appear to attentively discriminate over, by detecting high weights to emphasize both texture and shape features with five-digit precision as opposed to the color

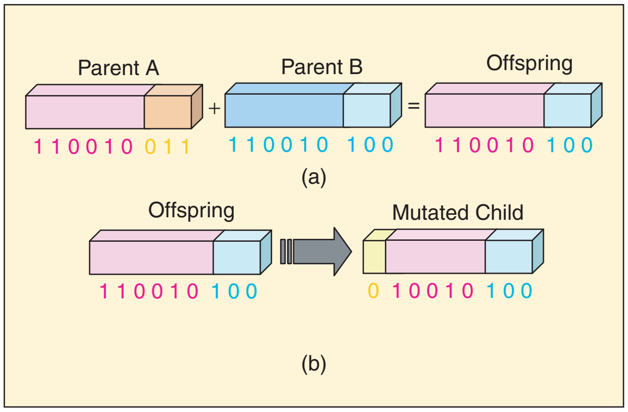


FIGURE 10 Genetic operators: (a) one-point crossover operation and (b) gene mutation operation.

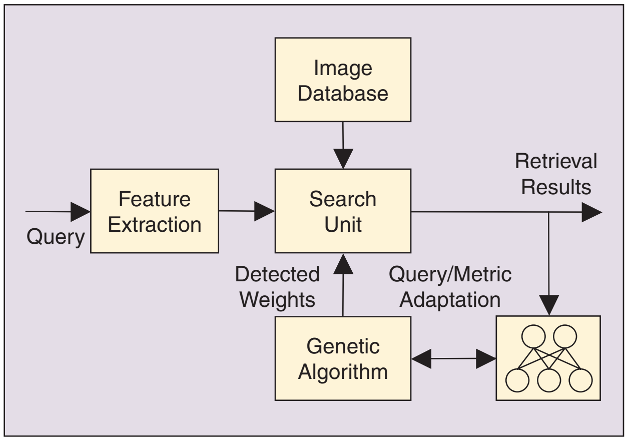


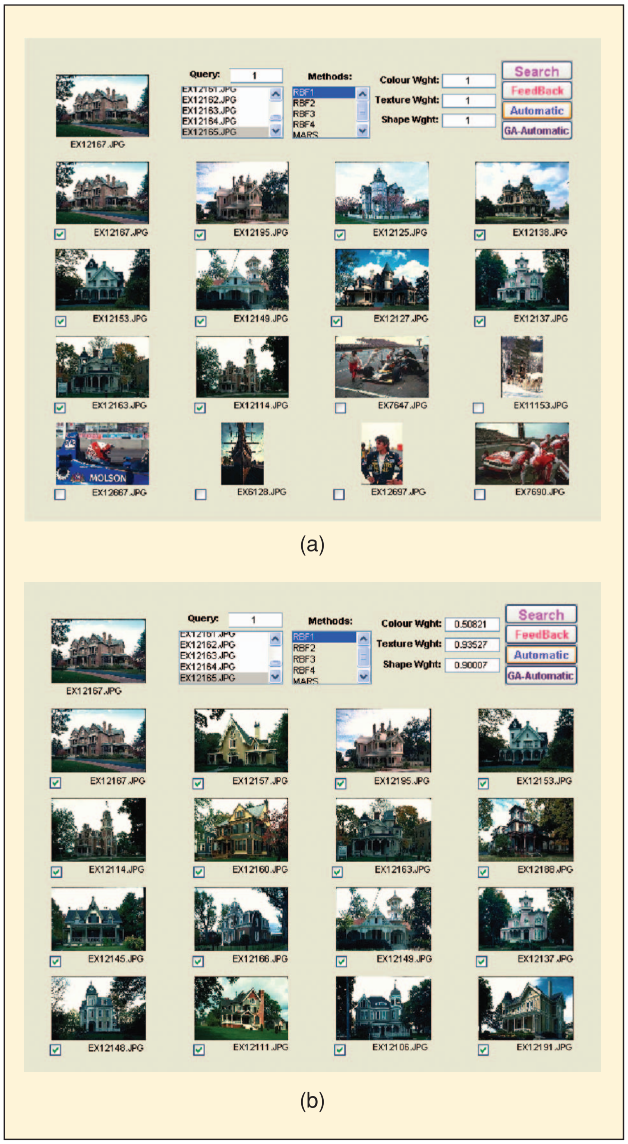
FIGURE 11 Genetic algorithm-based CBIR.

features, which have a moderate contribution to the DSOTM classification.

VI. SOTM-Cluster Modeling for Microbiological Image Analysis

Microscopy has long been considered an essential investigative tool in the biological sciences. Recent advances in *confocal* modalities, in particular, have allowed for the non-invasive capture of spatially registered 3-D information regarding the physical and chemical structure of complex microbiological environments. With possibilities for new insights, interest is shifting from how to collect the *pretty picture* toward the more significant issue of how to adequately quantify this information for more systematic visualization, characterization and modeling.

In such scenarios, any form of evaluation would typically be guided by an expert. Analysis often proceeds with an initial segmentation, wherein the expert attempts to identify meaningful *biomass* he/she wishes to analyze. Such steps become quite prohibitive if dealing with large 3-D volumes of data. In addition, they are rarely reproducible, particularly if evaluating data sets are highly heterogeneous (a common feature of biological image data). As such, an unacceptable level of subjectivity is introduced into the analysis.


(a)

Query: 1

Methods: RBF1

Colour Wght: 1

Texture Wght: 1

Shape Wght: 1

(b)

Query: 1

Methods: RBF1

Colour Wght: 0.50821

Texture Wght: 0.93527

Shape Wght: 0.90007

FIGURE 12 Screenshot of the GA-CBIR Engine: (a) 62.5% retrieval rate for the query image (on top-left corner) using DSOTM classifier without GA feature weight detection; and (b) 100% retrieval rate for the same query using GA feature weight detection and DSOTM classifier.

In biofilm research, for example, unsupervised approaches based on SOTM technology [24], [28] have been used to automatically formulate topologically-structured maps of dominant voxel patterns exhibited in collected 3-D image datasets. The maps then serve as a backbone description of the data from which associated constituents may be extracted for further evaluation.

The advantage of having a topological breakdown based on the SOTM in this application is twofold. First, the SOTM in hierarchically parsing the dataset allocates prototypes to prioritize maximal discrimination in the data. Second, discovery of inter-relationships among classes that form can be utilized for automating visualization and in higher level characterization of heterogeneous data. A system diagram is shown in Figure 13.

A. Visualization

As a visualization example, a volume rendering of a dataset could use the associations among classes to establish how to assign transparencies and color schemes to enhance or focus visualization on particular regions (or constituents) of interest.

Figure 14 shows two such examples. The first is in the extraction of orchid root tip chromosomes from a confocal stack captured using a phase-contrast modality. The modality, known as Nomarski Differential Interference Contrast (DIC)_[40], is necessary as the chromosomes are translucent in their natural state. Often, such methods yield subtleties in

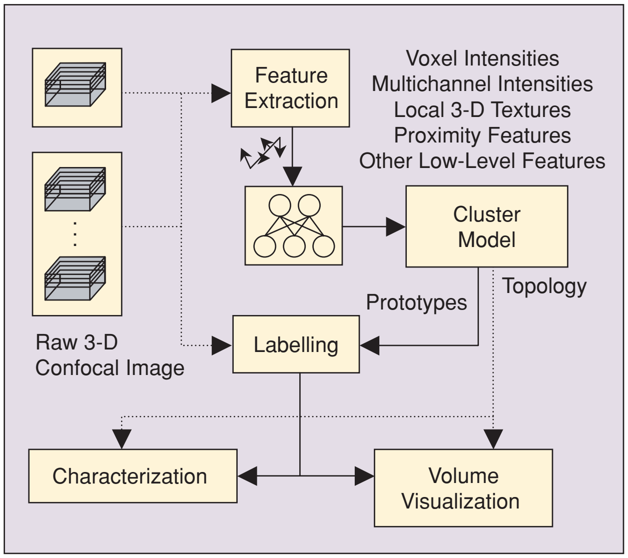


FIGURE 13 Systematic approach to microbiological image analysis. SOTM forms a cluster model, which acts as a backbone for visualisation and characterization tasks.

structural variation that cannot be detected with an epi-fluorescent modality (which on the other hand, might be capable of targeting a specific molecular constituent). In the current treatment, visualization of the chromosomes is achieved through segmentation with the SOTM. This is an extension of research presented in [41], where an SOM methodology was adopted. In the SOTM case, the transition into a volume visualization framework is more natural, as topological associations provide a more suitable associative backbone through which opacities of extracted microstructure types may be more readily controlled. In this instance, two dominant voxel prototypes within the chromosome are isolated and emphasized with imposed color contrast. Figure 14(a) shows a sample chromosome slice and the resulting volume-rendered dataset.

A second visualization example can be seen in an application to a biofilm sample that has grown on a filter membrane. Such biofilm communities present a problem in water treatment systems, as they eventually foul the membrane on which they grow.

Biofilm are essentially aggregations of complex bacterial communities with organic and non-organic materials that adhere to a substrate or form as floc, suspended in aqueous environments [42]. They are of *major* interest in environmental and medical industries due to their inherent resilience to environmental stresses. They remain the major cause of about 80% of infection in humans. Alternatively, in water treatment, the ability to control and predict their growth can be exploited to assist in the extraction of pollutants and toxins.

In Figure 14(b), the dataset is multi-channelled, and represents the collection of spatially-registered responses to three fluorescent markers, each targeting a specific DNA component. The goal was to utilize the discriminative and adaptive power of the SOTM to extract dominant mixtures (voxel patterns) to formulate a model of the bacterial communities existing on the membrane and the membrane itself. The result was the automatic extraction of the biofilm constituents. The membrane has been rendered semi-transparent for context. These images were rendered using AmiraDev 1.3 software [43]. Traditionally, to achieve such segmentation requires

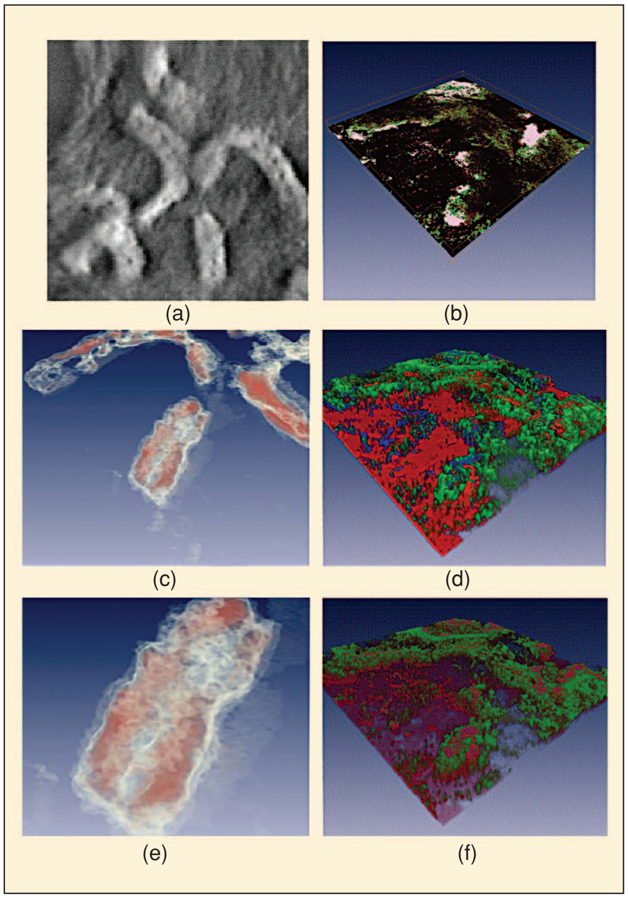


FIGURE 14 SOTM segmentation for volume visualisation. Chromosome visualisation: (a) original slice 9/18, (c) and (e) segmented and volume-rendered. Biofouling segmentation: (b) original slice, (d) rendering of three channels artificially colored, and (f) segmented biofilm layer—bacterial communities isolated from filter membrane and one another; filter membrane semi-transparent for context.

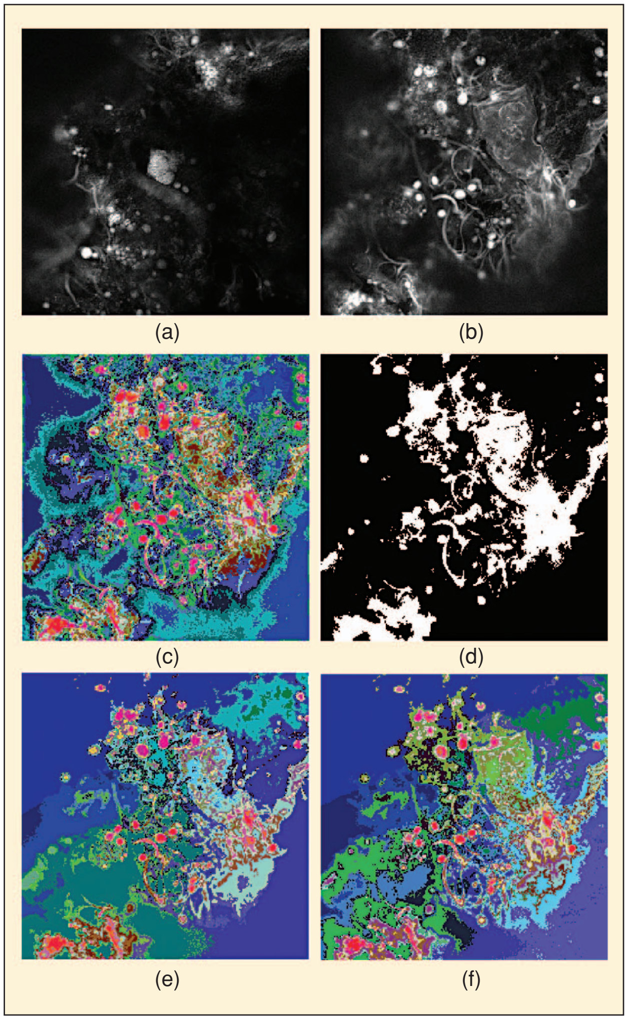


FIGURE 15 SOTM segmentations of biofilm: (a),(b) two slices of biofilm. Segmentations of (b): (c) K-means and (d) Otsu's threshold; (e) SOTM; and (f) SOTM with dynamic feature weighting during learning. The final two cases show increased structural resolution and association in a manner that appears more perceptually relevant.

much user effort in the manual adjustment of thresholds, pixel selection, etc.—within the framework of the program.

B. Characterization and Analysis

In another example, the problem of user-driven subjectivity for characterization is highlighted. Consider two slices from different fields of view within a biofloc specimen (Figure 15 (a), (b)). Visually, they appear quite different; however, there are aspects that are similar; for instance, the localized clusters of bacteria nodules forming in surrounding fine grain material.

Typically, characterization processes would involve some course grain thresholding, or automated thresholding such as the use of Otsu's method [44] or ICM [45]. Such methods form a precursor to characterization software such as ISA [46]—an industry standard product for quantifying biofilm. In cases such as Biofloc, the bio-mass present is of a highly heterogeneous nature, thus much potentially important information about characterizing the state of the biofilm/floc is lost. This is indicated in Figure 15(d), where Otsu's threshold cannot discriminate between internal constituents from Figure 15(b).

The well-known K-Means algorithm was used in Figure 15(c), initialized with the same number of classes as discovered by the SOTM result in Figure 15(e). In this case, proximity features were incorporated in the feature extraction phase. The SOTM demonstrates a highly efficient allocation of classes to foreground *Biomass*. It represents a much more perceptually relevant decomposition of the original image slice of Figure 15(b) than was achieved with K-Means. Its growth in this case was limited by imposing a data-dependent lower limit on the hierarchical control function. Interestingly, relationships associating nearby particles of similar content have also been isolated by the SOTM. This hierarchical grouping is evident somewhat in the two clustered sets of bacterial nodules forming two major spatial groupings seen in Figure 15(e).

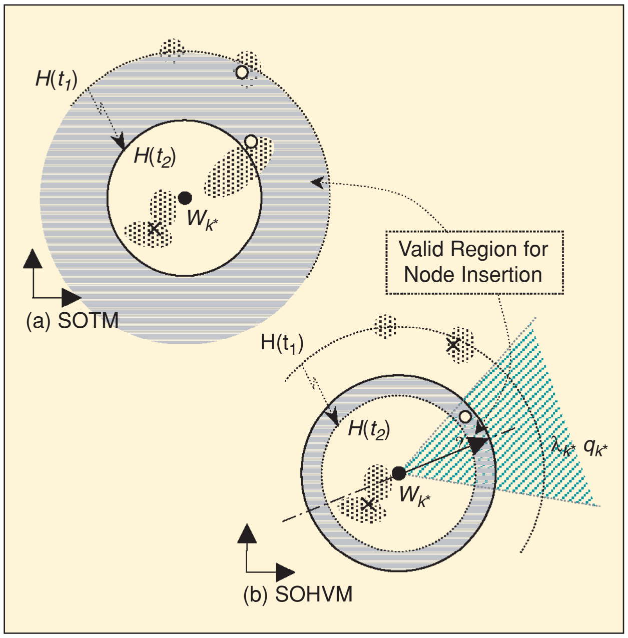


FIGURE 16 A comparison of the decision regions used for node generation in an SOTM versus an SOHVM. The latter attempts to new insert nodes (open circles) at the extremities of the data described by a winning node (closed circle) [w_{k^*} denotes the winning center, and its eigenvalue/vector pair describe local variance properties and is denoted by (λ_{k^*}, q_{k^*})]. X's are invalid sites for node insertion.

The final image, Figure 15(f), shows an additional experimental result that implements a dynamic feature weighting strategy in the process of learning. Essentially, proximity features were favored early during learning, and signal features were favored later. The flexibility of the SOTM tree structure allowed for an easy shift in allocation across these two feature subspaces. In fact, along with the hierarchical parsing, this worked to resolve even more structural subtleties as discrimination was maximized at all levels of evolution. This was also suggestive of a possible joint mechanism that might be at play when the natural process of segmentation of unknown visual content takes place in the human visual system. At any rate, the hierarchical discriminative property embedded into the SOTM process of self-organization does present some interesting perceptual abilities.

Final groupings formulated in the SOTM network and its structure may be exploited for inferring higher level properties, thus leading to more pure characterization potential within such architectures. In considering datasets collected from sets of similar biofilm sub-samples, *identifying* characteristics could in fact manifest in the associations formed by the network. Such characteristics could also then be correlated with other biological measurements taken outside the scope of imaging. This could lead to improved models of such biological systems, which in turn promotes understanding of behavior, management and control.

VII. Local Variance Driven Self-Organization: Toward the Generation of Forests from Trees

A novel unsupervised architecture known as the Self-Organizing Hierarchical Variance Map (SOHVM) has recently been proposed [47], and is also rooted in the principles of the SOTM. This model differs from the SOTM process in that it utilizes an *intelligent* vigilance strategy based on the underlying data distribution as it is discovered, rather than a globally decaying hierarchical function.

In addition to SOTM-based parent/children relationships, CHL is incorporated into the SOHVM's adaptive mechanism. The methodology thus shares properties common to GNG variants in that, as it partitions the input space divisively, a topological preserving set of associative connections among nodes is simultaneously constructed. In doing so, it is also possible for regions of the data space to disassociate at lower hierarchical levels. In this sense, the model begins with the growth of *trees* but may expand at later phases into *forests of trees* encoding even higher levels of association.

Each neuron in the network consists of a dual memory element that tracks information regarding a discovered prototype. In addition to center position (as is the case with most other self-organizing frameworks), Hebbian-based Maximum Eigenfilters (HME) [48], simultaneously estimate the maximal variance of local data. Such units present an additional realization of the Hebbian postulate of learning (Section I) and essentially probe online the nature of the covariance in the data within the vicinity of any given node. The HME module embedded in each node provides for *local orientation selectivity* of the underlying data, as the HME extracts the maximal eigenvector/eigenvalue pair from the locally surveyed data.

Vigilance is now assessed via interplay among local variances extracted such that more informed decisions control and naturally limit network growth. Figure 16 shows a comparison between the decision regions that result in the insertion of a new node in the SOTM versus the SOHVM. The hierarchical threshold begins as in the SOTM (very large), yet as nodes are inserted, regions of the data are isolated, thus their local variances are generally reduced. The hierarchical threshold decays in accordance with the variances discovered in the network. This continues as long as reasonable clusters are extracted. In regions where the hierarchical function falls below the local variance, the node is disabled, such that it can no longer act as a site for the spawning of new children. This limits over-fitting.

Once the dense regions have been exposed, the network attempts to insert nodes into outlier or noise positions from which it is difficult to track back due to increased competition across the data space. In the SOHVM, however, the discovery of such an outlier (typified by high local variance and low probability) also drives the local variance high, disabling the node. Ultimately, the SOHVM intelligently decides when to cease growing, thereby addressing one of the most difficult problems in unsupervised clustering—inferring an appropriate number of clusters in which to partition the data.

Figure 17 gives an indication of how the network unfolds over a sample data space (in comparison to GNG). In Figure 18(b), complete disassociation occurs between the upper left and lower right portions of the data, thus producing multiple trees (*a forest of trees*) encoding even higher levels of association. In the noisy example of Figure 18(d), an outlier has helped to shut down the production of nodes, thus naturally limiting the final configuration.

VIII. Closing Remarks

Complex tasks in multimedia processing that normally require extensive human interaction or guidance become prohibitive when there are overwhelming volumes of data

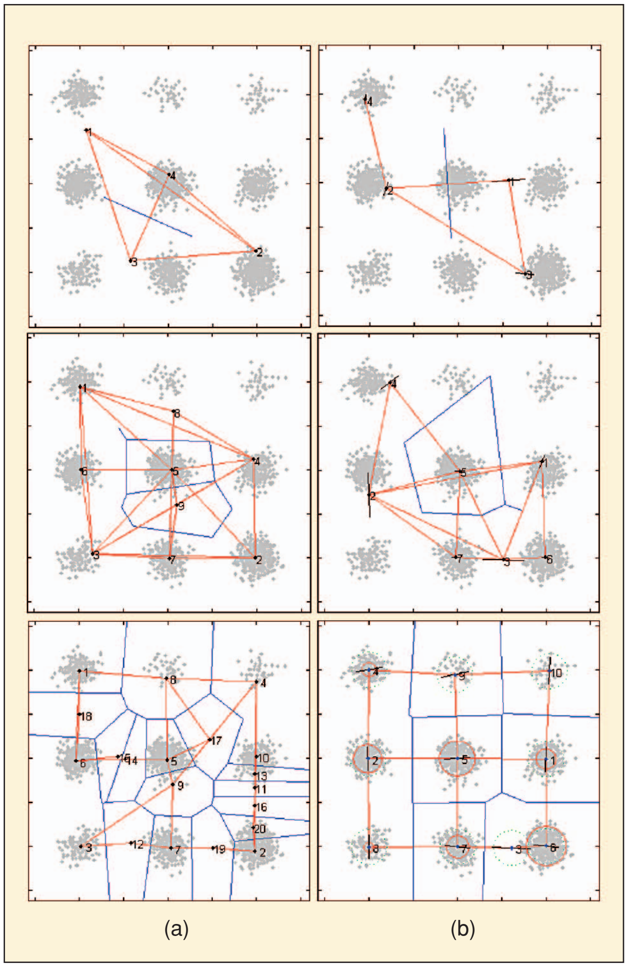


FIGURE 17 (a) GNG evolves onto data space, but splits prematurely well before capturing all cluster positions. Classification thus subdivides clusters. (b) SOHVM maximally spans the data space through hierarchical partitioning; this property is localized through the HME units (lines show axis for orientation selectivity). Network infers more accurate number of total clusters.

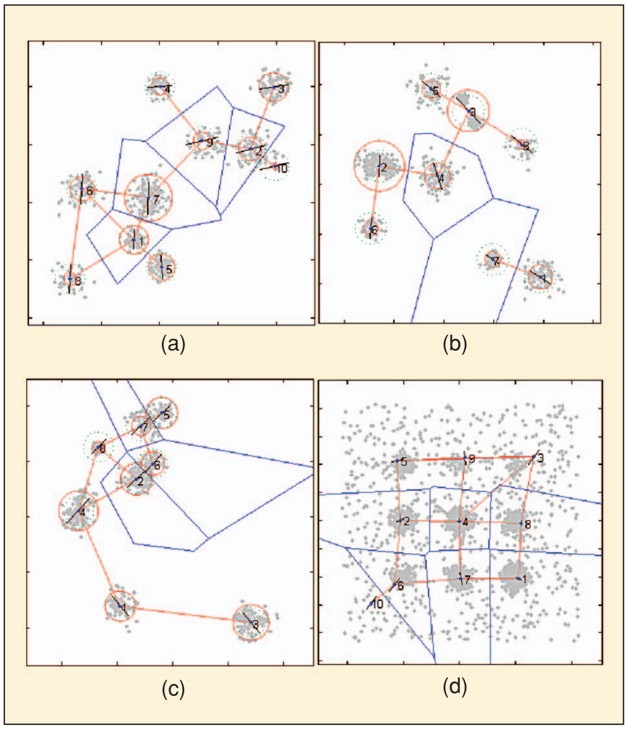


FIGURE 18 Sample clustering on 2-D datasets using variously sized, spaced and density clusters. (a)–(c): In (b), two trees form (*forest*) as the content in the lower region becomes disassociated from the content in the top, representing an even higher level of association. (d) shows data from Figure 17, with noise; outlier triggers stop. Connections among nodes represent final associations.

that need to be processed. In most, a certain amount of perception is required to successfully complete them (a highly non-linear challenge). Artificial neural networks based on self-organization are particularly suited to modeling such nonlinearities, thus they have generated much research interest. Dynamically growing models of self-organization attempt to address the problem of inferring an appropriate number of classes from the underlying data distribution, and thus, exhibit greater flexibility than static approaches. In this discussion, the SOTM family of architectures is demonstrated for their propensity in formulating a topological set of classes that attempt to maximize pattern discrimination while hierarchically parsing an underlying distribution. The SOTM along with newer variants (DSOTM and SOHVM) have been proposed and applied with success to automatic content-based image retrieval, and to the segmentation and analysis of 3-D microbiological image data. In these examples, SOTM variants prove to be an imperative component in the automation of perceptually-driven tasks. With the growing demand for such automation by many of today's advanced multimedia applications, dynamic self-organization seems well-poised to leave its mark.

Acknowledgments

The authors would like to thank the *Natural Sciences and Engineering Research Council of Canada* (NSERC) for supporting Kambiz Jarrah; the *Canada Research Chair* (CRC) program for supporting Matthew Kyan; *Canada Foundation for Innovation* (CFI) and Ryerson University for infrastructure support; and the *Environmental Biotechnology Laboratory* at Ryerson University, in particular Steven Liss, Ivan Zhu and Heather Kraemer for supplying biofilm data and their continuing support and expertise.

References

[1] A.M. Turing, "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society, B*, vol. 237, pp. 5–72, 1952.

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed., Prentice Hall, New Jersey, 1999.

[3] D.O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, New York: Wiley, 1949.

[4] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69.

[5] G.G. Yen and Z. Wu, "Ranked centroid projection: A data visualization approach for self-organizing maps," *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, pp. 1587–1592, July 31–Aug. 4, 2005.

[6] T. Kohonen, "Self-organization and associative memory," Springer-Verlag, Berlin, 1984.

[7] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, pp. 121–131, 1976.

[8] T. Martinetz and K. Schulten, "A 'Neural-Gas' network learns topologies," *Artificial Neural Network*, vol. I, pp. 397–402, 1991.

[9] R. Miikkulainen, "Script recognition with hierarchical feature maps," *Connection Science*, vol. 2, 1990.

[10] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, S. Amari, C. L. Giles, M. Gori, and V. Puri, Eds., vol. 6. Como, Italy: IEEE Computer Society, pp. 15–19, July 24–27, 2000.

[11] M. Dittenbach, A. Rauber, and D. Merkl, "Recent advances with the growing hierarchical self-organizing map," in *Advances in Self-Organizing Maps*, N. Allinson, H. Yin, L. Allinson, and J. Slack, Eds. Lincoln, GB: Springer-Verlag, pp. 140–145, June 13–15, 2001.

[12] M. Dittenbach, A. Rauber, and D. Merkl, "Uncovering the hierarchical structure in data using the growing hierarchical self-organizing map," *Neurocomputing*, vol. 48, no. 1–4, pp. 199–216, Nov. 2002.

[13] M. Dittenbach, D. Merkl, and A. Rauber, "Hierarchical clustering of document archives with the growing hierarchical self-organizing map," *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2001)*, Vienna, Austria, Aug. 21–25, 2001.

[14] P. Koikkalainen and E. Oja, "Self-organizing hierarchical feature maps," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 1990)*, San Diego, CA 1990.

[15] J. Pakkanen, J. Iivarinen, and E. Oja, "The Evolving Tree—A novel self-organizing network for data analysis," *Neural Processing Letters*, vol. 20, no. 3, pp. 199–211, 2004.

[16] B. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.

[17] B. Fritzke, "Growing Grid—A self-organizing network with constant neighborhood range and adaption strength," *Neural Processing Letters*, vol. 2, no. 5, 1995.

[18] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems* 7, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. MIT Press, pp. 625–632, 1995.

[19] T.M. Martinez, "Competitive Hebbian learning rule forms perfectly topology preserving maps," In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 1993)*, Amsterdam, Netherlands, pp. 427–434.

[20] D. DeSieno, "Adding a conscience to competitive learning," *IEEE International Conference on Neural Networks*, San Diego, CA, vol. I, pp. 117–124, 1988.

[21] C. Hung and S. Wermter, "A dynamic adaptive self-organizing hybrid model for text clustering," *Third IEEE International Conference on Data Mining (ICDM 2003)*, p. 75.

[22] H. Kong and L. Guan, "Detection and removal of impulse noise by a neural network guided adaptive median filter," *Proc. IEEE Int. Conf. on Neural Networks*, pp. 845–849, Perth, Australia, 1995.

[23] J. Randall, L. Guan, X. Zhang, and W. Li, "Investigations of the self-organizing tree map," *Proc. Of Int. Conf. on Neural Information Processing*, vol 2, pp. 724–728, Nov. 1999.

[24] M. Kyan, L. Guan and S. Liss, "Dynamic feature fusion in the self-organizing tree map—applied to the segmentation of biofilm images," *Proc. Of Int. Joint Conf. on Neural Networks*, Montreal, Canada, July 31–Aug. 4, 2005.

[25] S. Grossberg, *Studies of Ming and Brain*, Boston: Reidel, 1982.

[26] G. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," *Computer*, vol. 21, no. 3, pp. 77–87, 1988.

[27] G.A. Carpenter, S. Grossberg, and J.H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991.

[28] M. Kyan, L. Guan, and S. Liss, "Refining competition in the self-organizing tree map for unsupervised biofilm segmentation," *Neural Networks*, Elsevier, vol. 18, no. 5–6, pp. 850–860, July–Aug. 2005.

[29] P. Muneesawang and L. Guan, "Interactive CBIR using RBF-based relevance feedback for WT/VQ coded images," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Salt Lake City*, USA, pp. 1641–1644, vol. 3, May 2001.

[30] P. Muneesawang, "Retrieval of image/video content by adaptive machine and user interaction," Ph.D. Thesis, The University of Sydney, 2002.

[31] P. Muneesawang and L. Guan, "Automatic machine interactions for content-based image retrieval using a self-organizing tree map architecture," *IEEE Trans. on Neural Networks*, vol. 13, no. 4, pp. 821–834, July 2002.

[32] P. Muneesawang and L. Guan, "Using knowledge of the region of interest (ROI) in automatic image retrieval learning," *International Joint Conference on Neural Networks (IJCNN05)*, Montreal, Canada, Aug. 2005.

[33] Y. Rui and T.-S. Huang, "Optimizing learning in image retrieval," in *Proc. Of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hilton Head, SC, vol. 1, pp. 236–243, June 2000.

[34] K. Jarrah, M. Kyan, S. Krishnan, and L. Guan, "Computational intelligence techniques and their applications in content-based image retrieval," *IEEE Int. Conf. on Multimedia & Expo (ICME)*, submitted for publication.

[35] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, 1989.

[36] W.R. Smart and M. Zhang, "Classification Strategies for Image Classification in Genetic Algorithm," *IEEE Image and Vision Computing NZ*, Palmerston North, pp. 402–407, Nov. 2003.

[37] L.I. Kuncheva and L.C. Jain, "Designing classifier fusion systems by genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 327–336, Nov. 2000.

[38] J. Yang, V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intelligent Systems*, vol. 13, no. 2, pp. 44–49, Mar.–Apr. 1998.

[39] S. Zehang, Y. Xiaojing, G. Bebis, and S.J. Louis, "Neural-network-based gender classification using genetic search for Eigen-Feature selection," *Proceedings of the 2002 International Joint Conference on Neural Networks.*

[40] C.J. Cogswell and C.J.R. Sheppard, "Confocal differential interference contrast (DIC) microscopy: Including a theoretical analysis of conventional and confocal DIC imaging," *Journal of Microscopy*, vol. 165, no. 1, pp. 81–101, 1992.

[41] M.J. Kyan, L. Guan, M.R. Arnison, and C.J. Cogswell, "Feature Extraction of Chromosomes From 3-D Confocal Microscope Images," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 11, pp. 1306–1318, Nov. 2001.

[42] S.N. Liss, "Microbial flocs suspended biofilms," In G. Sutton (Ed.), *Encyclopedia of Environ. Microbiology*, Wiley & Sons, NY, pp. 2000–2012, 2002.

[43] AmiraDev 1.3, Mercury Computer Systems, www.tgs.com

[44] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. On Systems Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[45] J. Besag, "On the statistical analysis of dirty pictures (with discussion)," *J. Roy Statist Soc Ser B.*, vol. 48, pp. 259–302, 1986.

[46] Image Structure Analyzer ISA, Center for Biolfilm Engineering, Montana State University.

[47] Local variance driven self-organization for unsupervised clustering, *submitted to International Conference on Pattern Recognition (ICPR)*, Hong Kong, Aug. 20–24, 2006.

[48] E. Oja, "A simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, vol. 15, pp. 267–273.