# A Cellular Coevolutionary Algorithm for Image Segmentation

Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer, *Member, IEEE*

***Abstract*—Clustering is inherently a difficult problem, both with respect to the definition of adequate models as well as to the optimization of the models. In this paper we present a model for the cluster problem that does not need knowledge about the number of clusters *a priori*. This property is among others useful in the image segmentation domain, which we especially address. Further, we propose a cellular coevolutionary algorithm for the optimization of the model. Within this scheme multiple agents are placed in a regular two-dimensional (2-D) grid representing the image, which imposes neighboring relations on them. The agents cooperatively consider pixel migration from one agent to the other in order to improve the homogeneity of the ensemble of the image regions they represent. If the union of the regions of neighboring agents is homogeneous then the agents form alliances. On the other hand, if an agent discovers a deviant subject, it isolates the subject. In the experiments we show the effectiveness of the proposed method and compare it to other segmentation algorithms. The efficiency can easily be improved by exploiting the intrinsic parallelism of the proposed method.**

***Index Terms*—Clustering, distributed genetic algorithms, image segmentation, modeling.**

## I. Introduction

CLUSTERING is an important and difficult task in unsupervised pattern recognition. The clustering problem comes down to finding the separation of a set of objects into an *a priori* unknown number of subsets while minimizing intra-cluster variability (within scatter) and maximizing the inter-cluster variability (between scatter). There is a huge amount of literature on the subject, ranging from models, algorithms, algorithm parameter estimations to cluster validity studies [14], [33]. The clustering methods can be divided up into exclusive and nonexclusive methods [26]. The best-known nonexclusive method is the fuzzy C-means model [15]. In this method objects are soft clustered such that objects belong to all clusters to a certain degree. For an overview of fuzzy clustering methods see, for example, [3] and [4]. In exclusive clustering methods, the objects are partitioned into a number of (crisp) subsets, such that each object belongs to exactly one subset. We concentrate on exclusive clustering methods, among which the K-means model (or hard C-means) [28] is the most widely used. As the fuzzy C-means model, the K-means model assumes that the number of clusters is known *a priori*. There are, however, numerous domains for which this assumption cannot be satisfied. One of these is the image segmentation problem, which we especially address in this paper. In (region-based) image segmentation, pixels are clustered based on their color or texture information, while a hard constraint is imposed on spatial cluster (segment) connectivity. Throughout this paper, we will consider the clustering problem and the segmentation problem as being similar. Accordingly, we consider solution methods for both problems interchangeably.

Since the clustering problem is a known NP-hard problem, deterministic algorithms for the fuzzy C-means and the K-means model use a greedy optimization scheme in order to find a suboptimal solution of their criterion function. Many stochastic optimization schemes that aim at a global maximum have been reported, among which are simulated annealing methods [9], [24], [32] and evolutionary algorithms [12], [18]–[20], [27], [36]. As part of some evolutionary approaches also certain domain specific recombination operators have been reported [7], [8], [25], [35].

In this paper, we introduce a cluster model which aims at minimal intra-cluster variability. Additionally, instead of maximizing the inter-cluster variability we impose a hard constraint on the intra-cluster variability for the union of two clusters. We claim that such a constraint is inevitable in order find useful solutions to the cluster problem. As a result the proposed model allows for the clustering of a data set into an *a priori* unknown number of clusters. Additionally, we specialize the model for image segmentation and propose a *cellular coevolutionary algorithm* (CCA) to optimize the image segmentation model in a distributed way.

The outline of the paper is as follows. After first exploring the characteristics of the clustering problem, we propose a new cluster model in Section II. Then, in Section III, we specialize the model for image segmentation. In Section IV, we describe a coevolutionary algorithm for the optimization of the proposed model. In the experiments in Section V, we demonstrate the effectiveness of the method and compare its performance to some other image segmentation algorithms, both with synthetic and natural images. We finalize the paper with a discussion and some concluding remarks in Section VI.

## II. Cluster Model

In this section, we elaborate on some fundamental characteristics of the cluster problem. We will not consider the definition of an appropriate distance measure, which is a known problem especially when multi-variate features are involved. The characteristics that we focus on are manifest, irrespective of the used

Manuscript received November 20, 2001; revised August 13, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aleksandra Mojsilović.

The authors are with the Department of Mediamatics, Faculty of Information Technology and Systems, Delft University of Technology, 2600 GA, Delft, The Netherlands (e-mail: C.J.Veenman@its.tudelft.nl; M. J. T. Reinders@its.tudelft.nl; E.Backer@its.tudelft.nl).

Digital Object Identifier 10.1109/TIP.2002.806256

1057-7149/03$17.00 © 2003 IEEE

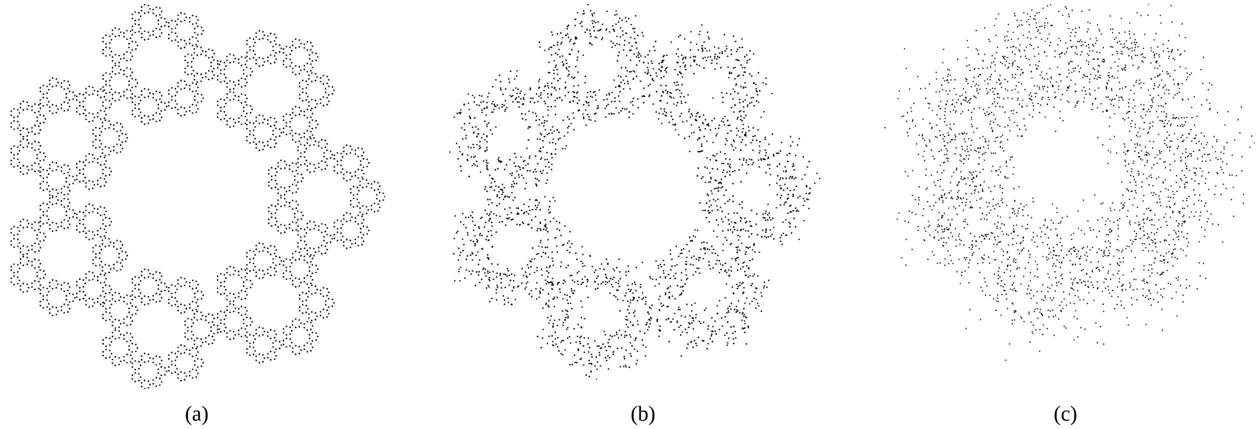


Fig. 1. (a) Noiseless 2-D data set consisting of a nested circular structure. In (b) and (c) the same data set is displayed with an increasing amount of Gaussian noise. The number of distinguishable clusters decreases accordingly.

distance measures. Before going into detail, we first define the clustering problem.

Given is a data set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, where $\boldsymbol{x}_i$ is a feature vector in a $p$-dimensional metric space, and $N = |X|$ is the number of objects in $X$. Then, a valid clustering of $X$ in a set of clusters $C = \{C_1, C_2, \ldots, C_M\}$, where $M$ is the number of clusters, has the following partition properties:

P-1. $C_i \neq \emptyset$, $1 \leq i \leq M$;
P-2. $\bigcup_{i=1}^{M} C_i = X$;
P-3. $C_i \cap C_j = \emptyset$, $i \neq j$, $1 \leq i, j \leq M$.

Additionally, the clusters should reflect the structure of the data such that objects in the same cluster are similar to each other and objects from distinct clusters are different from each other. In order to find a solution to the clustering problem, we need a quantitative way to distinguish between similar and dissimilar objects or, in other words, we need to quantitatively differentiate between homogeneous and inhomogeneous sets of objects. In the literature various alternatives have been reported to approach this task [14], [23], [33].

A common criterion to quantify cluster homogeneity is the sum-of-squared-error criterion

$$\sum_{i=1}^{M} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \mu(C_i)\|^2 \tag{1}$$

where

$$\mu(Y) = \frac{1}{|Y|} \sum_{\boldsymbol{x} \in Y} \boldsymbol{x}.$$

Without additional constraints (1) has a zero minimum for $M = |X|$. Therefore, in addition to minimizing this criterion, many cluster models, like the K-means and fuzzy C-means model, assume the number of clusters $M$ to be known beforehand. There is, however, also a number of methods that among others assume a certain maximum variability per cluster, such as the split-and-merge and region growing algorithms in the image segmentation domain. We claim that objective clustering is not possible without such additional parameters because of a *scale problem*. We use Fig. 1(a) to support this observation. It is impossible to decide which is the right data clustering for the data set displayed in the figure: should this data set be partitioned into 1, 7, 49, or even more clusters? For synthetic data sets with self-similar structures like those displayed here, the problem is clearly undecidable. This may seem an academic problem that does not correspond to real life, however, if we consider the image in Fig. 2, we also see a nesting of structures. Again the number of clusters (segments) is arguable. For example, does the image only contain a tree, houses, garden, and sky, or must a valid segmentation result also include clouds, bushes, windows, or even smaller segments/clusters like the flowers and the leaves of the tree?

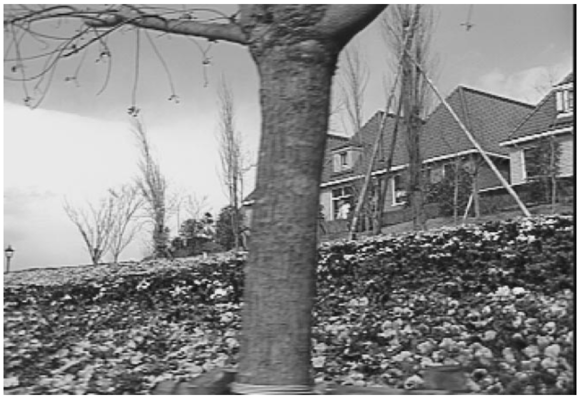


Fig. 2. Image from the Flower-Garden sequence. Also in this natural image the number of segments is clearly arguable.

A typical approach to discover a significant set of clusters is to minimize (1) for a range of settings of $M$, where $1 \leq M \leq M_{\max}$. Then, a cluster validity study [6], [11], [16], [22] can help in selecting the "true" number of clusters by looking for sharp knees or local minima in a cluster validity index function curve, e.g., the Davies–Bouldin index [11]. However, these index functions can have multiple local minima and knees, or they sometimes contain no significant transition at all. Moreover, the deepest local minimum or the sharpest knee are by no means indications of the scale the user expects.

In addition to the scale problem, in many practical situations the features of the objects are noisy. Because the intra-cluster variability of the data increases proportionally with the amount of noise, the determination of the true cluster borders can be severely hampered, leading to a *noise problem*. In some cases the noise can be reduced by filtering, but the uncertainty principle [34] is a limiting factor. That is, by reducing the noise on the features we also decrease the accurateness of the estimated cluster border.

Both the scale and the noise problem should be properly handled in a clustering method in order to find objectively good cluster results. Both aspects are interrelated, since dealing with one problem affects the other. That is, the smallest possible scale depends on the effective noise level. Usually, both problems are implicitly attacked in the same way, which implies that the assumed optimal scale is the one just above the noise level. We illustrate this with the data sets shown in Fig. 1(b) and (c). If noise is considered as the limiting factor, then Fig. 1(b) contains 7 clusters and Fig. 1(c), which has even more noise, contains only one cluster. Once again, also in Fig. 1(b) and (c) the problem is undecidable, since the original data is the same as in Fig. 1(a).

In the image segmentation domain, in which we are especially interested, the number of clusters is not known *a priori*. Therefore, we do not want to fix the number of clusters in our model. However, as already mentioned, both straight minimization of the intra-cluster variability and maximization of the inter-cluster variability lead to undesirable trivial solutions, being $N$ clusters or 1 cluster, respectively. We choose to minimize the intra-cluster variability while at the same time constraining the intra-cluster variability of the union of two clusters. In this way, this intra-cluster variability constraint defines the scale at which two clusters can be differentiated from each other.

To minimize the intra-cluster variability, we use the sum-of-squared-error criterion (1). Here, we rewrite (1) such that it fits better to the model and the presented algorithm in the remainder. Further, we implement the joint intra-cluster variability constraint as a minimum variance for the union of two clusters. This leads to the following cluster model:

$$\min_{C} \sum_{C_i \in C} |C_i| Var(C_i), \tag{2}$$

where

$$Var(Y) = \frac{1}{|Y|} \sum_{\boldsymbol{x} \in Y} ||\boldsymbol{x} - \mu(Y)||^2 \tag{3}$$

subject to

$$\forall C_i,\, C_j,\, i \neq j \colon Var(C_i \cup C_j) \geq \sigma^2_{\max}. \tag{4}$$

In general, the optimization of this model leads to clusters having a variance below $\sigma^2_{\max}$. However, there are some rare situations in which the variance of individual clusters can exceed this limit.

*Proposition 1:* For the solution to the cluster model as defined in (2)-(4) the following holds: if there exists a cluster $C_a$ for which the variance exceeds the maximum variance limit $sigma^2_{\max}$, then all objects of $C_a$ have a distance smaller than $\sigma^2_{\max}$ to the center of cluster $C_a$ or they can not be separated from $C_a$ without violating the joint variance constraint in (4) with some cluster $C_b$. That is

$$\begin{aligned} Var(C_a) \geq \sigma^2_{\max} \Rightarrow \forall \boldsymbol{x} \in C_a \colon& \\ \big(||\boldsymbol{x} - \mu(C_a)||^2 < \sigma^2_{\max} \vee \exists\, C_b \colon& \\ Var(C_b \cup \{\boldsymbol{x}\}) < \sigma^2_{\max}\big)\,.& \end{aligned} \tag{5}$$

*Proof:* For the proof we use the inverse implication. So we prove that the variance of a cluster $C_a$ will not exceed $\sigma^2_{\max}$ if all remote cluster objects, that is, cluster objects for which:

$$||\boldsymbol{x} - \mu(C_a)||^2 \geq \sigma^2_{\max} \tag{6}$$

holds are outside the $\sigma^2_{\max}$ range of another cluster, i.e.:

$$\neg\left(\exists\, C_b \colon Var(C_b \cup \{\boldsymbol{x}\}) < \sigma^2_{\max}\right). \tag{7}$$

Let $C$ be the optimal clustering resulting from the minimization of (2) subject to (4) and $z = \sum_{C_i \in C} |C_i| Var(C_i)$. Further, assume that there is a cluster $C_a$ for which $Var(C_a) \geq \sigma^2_{\max}$. Then, $\exists\, \boldsymbol{x} \in C_a$ such that $Var(C_a - \{\boldsymbol{x}\}) < Var(C_a)$. Especially all objects $\boldsymbol{x}$ for which $||\boldsymbol{x} - \mu(C_a)||^2 \geq \sigma^2_{\max}$ are candidates. Let $C'_a = C_a - \{\boldsymbol{x}\}$ and $\{\boldsymbol{x}\}$ form two separate clusters. Now, still $Var(C'_a \cup \{\boldsymbol{x}\}) \geq \sigma^2_{\max}$ and since (7) applies, also in general $\forall\, C_i \colon Var(C_i \cup \{\boldsymbol{x}\}) \geq \sigma^2_{\max}$. However, since $Var(\{\boldsymbol{x}\}) = 0$ the minimum $z' < z$. This is contradictory to the assumptions, so given (6) and (7), there is *no* cluster $C_a$ for which $Var(C_a) \geq \sigma^2_{\max}$. ■

Informally, the implication of this proposition is that when a dispersed cluster is enclosed by compact clusters, these compact clusters may prevent the dispersed cluster from splitting. In other words, the presence of the compact clusters makes the cluster separation ambiguous since in that case, the homogeneity criterion (2) is at odds with the joint variability constraint (4).

Let us explain this in more detail. Suppose $C = C_a \cup C_b$ is an optimal clustering resulting from the optimization of (2)-(4). Then, generally both clusters will have a variance below $\sigma^2_{\max}$. Now suppose that the variance of cluster $C_a$ exceeds $\sigma^2_{\max}$ while $C_b$ is a compact cluster that is near to $C_a$ in feature space. Since $Var(C_a) > \sigma^2_{\max}$ one would expect $C_a$ to be split in two clusters $C'_a$ and $C''_a$ leading to a lower squared error while the joint variance constraint still holds for these two subclusters, i.e., $Var(C'_a \cup C''_a) \geq \sigma^2_{max}$. However, because of the presence of the compact cluster $C_b$ the joint variance constraint will be violated either for $C'_a$ or $C''_a$ with $C_b$. Ultimately, not even a single object $x$ can be separated from $C_a$ without violating the joint variance constraint with $C_b$, i.e., $Var(C_b \cup \{x\}) < \sigma^2_{max}$. Clearly, such a separation would lead to a lower squared error because a single object cluster has $Var(\{x\}) = 0$. Another way in which the variance of $C_a$ can be lowered is by moving a remote object $x$ from cluster $C_a$ to $C_b$. However, in the proposed case the squared-error criterion will increase because the distance between $x$ and the cluster center $\mu(C_a - \{x\})$ is smaller than between $x$ and the cluster center $\mu(C_b \cup \{x\})$. Concludingly, the variance of $C_a$ can neither be lowered by 1) separating an object because the joint variance constraint will be violated, nor be lowered by 2) moving an object to another cluster because the squared error will increase.

Since the cluster variance can both be caused by the distribution of the data and by the noise in the data acquisition process, knowledge about the noise can be exploited in this model. We elaborate on dealing with noise in Section IV, where we describe a specialized algorithm for image segmentation.

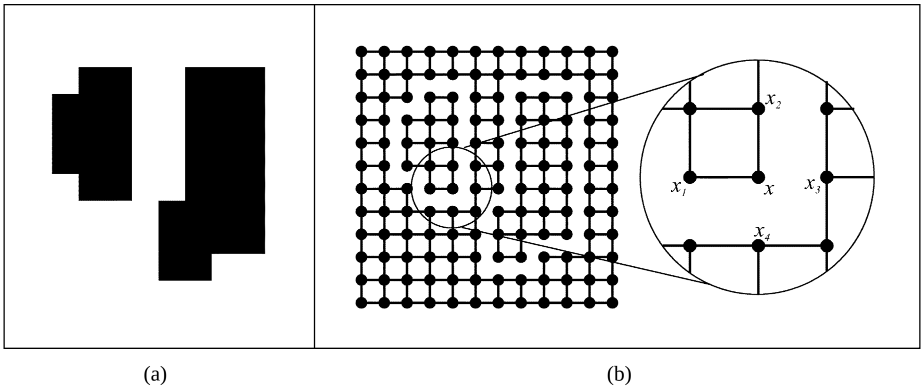


Fig. 3. (a) Image containing a background and two "objects," making three segments in total. (b) Corresponding segmentation graph and a blown up detail, where $\boldsymbol{x}$ has two adjacent vertices $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and four contiguous vertices, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, and $\boldsymbol{x}_4$.

## III. Image Segmentation Model

In this section we refine the just introduced cluster model for the image segmentation problem. The image segmentation problem is a special clustering problem where the objects are picture elements (pixels) and the feature vector consists of the pixel position information together with pixel appearance information.

The segmentation problem is defined differently from the cluster problem, since it requires that the segments are spatially connected. Several segmentation methods only impose a soft constraint on segment connectivity, e.g., [10], [17], [29], [30]. Accordingly, these methods are merely quantization methods, though the problems are similar to a certain extent. There are two additional reasons why we impose connectivity as a hard constraint. First, connected segments are usually bigger so that more *reliable statistics* can be maintained. Second, the optimization problem simplifies into a *one criterion optimization* problem, since we do not need to optimize a connectivity criterion as well. Moreover, we avoid the introduction of an additional parameter that is needed in case the resulting multi-criterion optimization problem is solved by weighting a homogeneity and a connectivity term.

We first define the image segmentation problem as a graph-partitioning problem. We represent the image as an undirected graph $G$, where the $N$ vertices (graph nodes) represent the appearance feature vectors of the pixels. For convenience we denote the vertices by the appearance feature vectors $\boldsymbol{x}_i$. The pixel position information is represented in the edges of the graph. Further, we define *contiguous vertices* as being contiguous on the image grid, so every vertex has exactly four contiguous vertices.[1] We write $\boldsymbol{x}_1 \rightleftharpoons \boldsymbol{x}_2$ if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are contiguous. Only if two vertices are contiguous, there can be an edge connecting them, which makes them adjacent. That is, every vertex can have at most four edges to contiguous vertices. In Fig. 3, we illustrate the relation between contiguous and adjacent vertices.

*Definition 1:* A *segmentation graph* is a graph for which the following holds: if there is a path between two vertices and the vertices are contiguous, then the vertices are adjacent, i.e., there is an edge between the vertices.

Every maximally connected subgraph (component) in a candidate segmentation graph is a candidate segment $S_i$, where $1 \leq i \leq M$ (and $M$ being the number of segments). A candidate segment $S_i$ has a set of vertices $C_i = V(S_i) = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_i}\}$, where $N_i$ is the number of vertices in $S_i$.

A consequence of the hard connectivity constraint is that (4) only has to be satisfied for *contiguous components*, where components $S_1$ and $S_2$ are contiguous: $S_1 \rightleftharpoons S_2$ if $\exists\, \boldsymbol{x}_1 \in C_1$, $\exists\, \boldsymbol{x}_2 \in C_2$: $\boldsymbol{x}_1 \rightleftharpoons \boldsymbol{x}_2$, with $C_1 = V(S_1)$ and $C_2 = V(S_2)$.

The objective of the segmentation task is to find a segmentation graph that represents homogeneous appearing regions. In this study we do not consider color images, so the regions must be homogeneous with respect to their intensities. In the general cluster model, we stated that a certain variability must be allowed for to establish a certain cluster scale. However, if the intensity feature is noisy the image variability caused by the noise must be incorporated too. So, in addition to the given intra-cluster variability to establish the desired scale, we assume that a variability is given due to noise. We measure both variabilities as pixel intensity variance present at the given scale ($\sigma_s^2$) or resulting from the amount of noise ($\sigma_n^2$), respectively. Then, $\sigma_{\max}^2 = \sigma_s^2 + \sigma_n^2$, since the underlying processes (signal and noise) can be assumed independent.

## IV. Algorithm

For the optimization of the cluster model, we propose a stochastic optimization algorithm. In the literature other stochastic clustering algorithms have been reported that generally optimize the K-means model or fuzzy C-means model either using simulated annealing techniques [9], [24], [32] or using evolutionary computation techniques [12], [18], [27], [36]. Accordingly, these stochastic approaches focus on the optimization of known cluster models. The algorithm we propose, however, shows more resemblance with the distributed genetic algorithm (DGA) for image segmentation as introduced by Andrey and Tarroux [1], [2]. We also dynamically apply local operators to gradually improve a set of hypothesized clusters. Important differences, however, are: 1) they do not consider the statistics of the whole segment in the optimization process, 2) their segments may become disconnected, which violates the strict definition of the segmentation problem, and 3) the

[1] If eight connectivity is desired, then a slightly different definition for contiguity should be applied.

selection operator that copies local model parameters and the segment label applies in a restricted neighborhood regardless of the color model similarities.

We now present the cellular coevolutionary algorithm (CCA) to optimize the model described in the previous sections. The proposed model has two important parameters, being the scale-derived image variance ($\sigma_s^2$) and the noise variance ($\sigma_n^2$). We deal with noise by simply pre-filtering the image with a uniform filter with size $f$. Consequently, the $\sigma^2_{\max}$ parameter from (4) is affected in the following way:

$$\sigma^2_{\max} = \sigma_s^2 + \frac{\sigma_n^2}{f} \tag{8}$$

since the total variance changes accordingly.

The proposed algorithm dynamically aims at optimizing (2), while satisfying (4) as well as the connectivity requirement. Accordingly, the algorithm has only one essential parameter, being $\sigma^2_{\max}$. The segments are maintained by active entities that autonomously try to optimize local criteria. We call these entities *agents* in order to stress their autonomous behavior. For efficiency reasons, each agent manages the outside border $B_i = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|B_i|}\}$ of the component it represents, where $|B_i|$ is the border length. This border is a list of vertices that are contiguous but *not* adjacent to the component the agent represents. In other words, $B_i$ contains candidate vertices with which the agent's component can be extended. In the following sections, we describe how the segments are initially created, how they evolve during the optimization process, and how the process terminates.

### A. Creation

Let the image have dimensions $W \times H$. Upon initialization the image is converted into a regular grid with $W \times H$ unconnected vertices, hence $N = W \times H$. Accordingly, the initial segmentation graph has as many components as vertices. Further, $N$ agents are created and each component is assigned to an agent. After creation the evolution process starts.

### B. Evolution

The evolution process is a sequence of epochs. During every epoch all agents (conceptually) act in parallel or alternatively sequentially in random order. An agent can take one of three actions. Alliances with neighboring agents are formed if the $\sigma^2_{\max}$ parameter allows it. The consequent component merge satisfies the scale condition (4). If no alliances are possible, an agent considers subject migration from a neighboring agent if this improves the global criterion (2). Now, no check is done whether the scale condition is violated. Afterwards (in a later epoch), subject isolation is considered, if in the course of subject migrations the variance exceeds $\sigma^2_{\max}$.

When called in sequential random order, agents are activated sequentially through a random wheel selection method, i.e., the most successful agents are more probable to be chosen again. An agent estimates its probability of success as the ratio of successful trials $s_i$ in the last $m$ trials. To avoid premature convergence, each agent, whether successful or not, has a basal chance (probability $r$) of being selected. The probability of an agent to be selected for activation is then given by

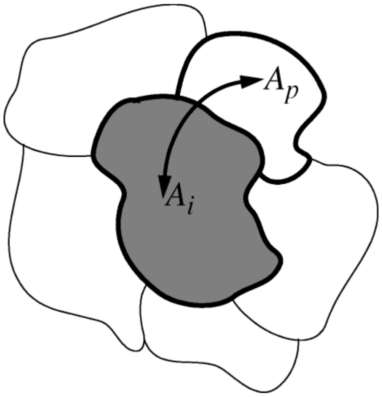


Fig. 4. Schematic illustration of alliance formation. The agent in the center is the one that attempts to form an alliance.

$$P_i = (1-r)\frac{s_i}{m} + r. \tag{9}$$

The agent will act in this epoch only if $P_i > U(0 \cdots 1)$, where $U(0 \cdots 1)$ is a uniformly distributed number in the range $[0 \cdots 1]$.

#### *Alliance Formation: Component Merge*

An agent first tries to form an alliance with any of its neighboring agents. Agents form alliances with their neighbors if the variance of the union of their components remains under the predefined maximum $\sigma^2_{\max}$ (4). Accordingly, for all $k_i$ neighbors the joint component variance is computed

$$\sigma^2_{ij} = Var(C_i \cup C_j). \tag{10}$$

Where the alliance that results in the lowest joint variance is selected. That is

$$p = \arg\min_{j=1}^{k_i} \sigma^2_{ij}. \tag{11}$$

If the variance of the selected alliance satisfies (4), then agent $A_i$ and $A_p$ become allies; see Fig. 4. We let $A_i$ represent the alliance by taking $A_p$'s component. Then, the neighboring agent $A_p$ no longer represents a graph component, so it terminates. Consequently, the number of segments $M$ decreases.

If the selected alliance violates (4), then the merge action fails for agent $A_i$ and the agent considers subject migration.

#### *Subject Migration: Vertex Exchange*

The aim of the subject migration is to contribute to the minimization of the global criterion (2). First, a number of candidate vertices are collected and the one that results in the highest gain is selected.

Agent $A_i$ collects $b_i$ candidate vertices from its border $B_i$ by random sampling; see also Fig. 5. It is important to note that at this point, the *greed* of the algorithm can be controlled. Considering all border objects ($b_i = |B_i|$) as candidates would make the algorithm very greedy, while taking very few objects restrains the optimization process.

Once the agent has collected a number of candidate vertices, it chooses that candidate vertex that delivers the highest positive

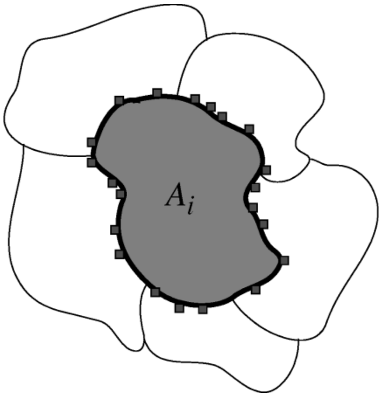


Fig. 5. Schematic illustration of subject migration. The agent in the center is collecting a number of candidates at its border.

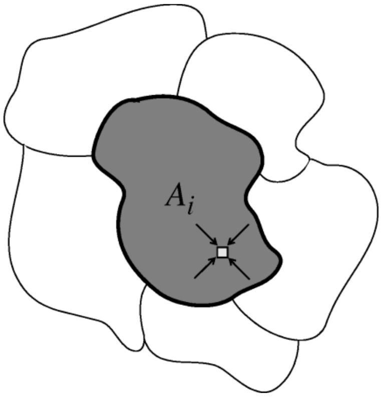


Fig. 6. Schematic illustration of subject isolation. The agent in the center selects the most deviant subject in case its component is too inhomogeneous.

gain with respect to the global criterion (2). To this end, the following local gain is computed for all candidates:

$$G_{ij} = g'_{ij} - g_{ij} \tag{12}$$

where $\boldsymbol{x} \in C_j$ and

$$\begin{aligned} g_{ij} &= |C_i|Var(C_i) + |C_j|Var(C_j) \\ g'_{ij} &= (|C_i|+1)Var(C_i \cup \{\boldsymbol{x}\}) + (|C_j|-1)Var(C_j - \{\boldsymbol{x}\}). \end{aligned} \tag{13}$$

If none of the candidates results in a positive gain, then the expansion trial fails since it does not contribute to the global criterion (2). Clearly, this is a *cooperative negotiation scheme* between agent $A_i$ and its neighboring agents.

Since segments must satisfy the connectivity constraint, vertex migrations from one segment to the other that violate this constraint must be recognized.[2] In other words, we have to check whether the candidate vertex $\boldsymbol{x} \in V(S_j)$ is a *cut vertex* of $S_j$. In the Appendix, we describe an efficient cut-vertex detection algorithm for this problem.

If the neighboring segment $S_j$ becomes disconnected as a result of the migration (because $\boldsymbol{x}$ was a cut vertex), one or two new agents are *created* that represent the disconnected components.

### *Subject Isolation: Vertex Removal*

Since the scale constraint was not checked during the subject migration, it can happen that the variance of the agents' component can exceed the given maximum $\sigma^2_{\max}$. In that case the agent decides to isolate the most deviant subject (see also Fig. 6). To this end, the agent first determines the vertex that has the largest (Eucledian) distance to the mean feature of the component. Then, a new agent is created to which this vertex is assigned. This action serves to satisfy the derived property in (5).

### *C. Termination*

For an algorithm to work in practical situations it has to terminate at a certain point. When no agent was able to successfully perform any of its actions for a certain number of epochs, we decide that the algorithm has converged. We have to wait for a number of failed epochs, because of the stochastic border sampling. We use the success counter $s_i$ for this purpose. So the algorithm terminates if $\forall A_i$: $s_i = 0$. Because the variance of a segment can indeed exceed $\sigma^2_{\max}$; see (5) local oscillations can occur. To escape from such situations the algorithm also stops when the agent community has evolved for a maximum number of epochs $E_{\max}$.

## V. Experiments

In order to evaluate the proposed image segmentation method, we performed two types of experiments. First, we explored some specific properties of the method to illustrate how it operates under various conditions and in what sense it can be adjusted. After that, we did some experiments in which we compared the CCA performance to other known image segmentation methods. In all experiments, we set the success ratio parameters (9) $m = 100$ and $r = 0.01$ and the border sampling parameter $b_i = \lfloor\sqrt{|B_i|}\rfloor$. These values are not very critical so we could fix them for all experiments. If not stated otherwise we set the filter size $f = 1$, which means that we did not pre-filter the images. We show the segmentation results in images, where the color of a segment equals the mean color of the pixels in the segment. Clearly the setting of $\sigma^2_{\max}$ is crucial for the final segmentation results. If not stated otherwise, an appropriate (optimal) level for $\sigma^2_{\max}$ was chosen. A similar procedure was followed for the methods with which we compared our approach (see Section V-B).

### *A. Exploring the CCA*

First, we show the course of the segmentation process with the Circle image in Fig. 7(a), in which the pixel intensity changes gradually. For this 128 × 128 image we set $\sigma^2_{\max} = 20^2$. In a strict sense this image can hardly be segmented, since there are no distinct regions to be separated. However, since gradually changing colors are quite often part of a natural scene, it is interesting to see how segmentation algorithms deal with such transitions.

Fig. 7(b)–(e) show some intermediate results of the segmentation process. In Fig. 7(c) it can be seen that after 10 epochs the final number of segments has been found. The remaining epochs

[2] Clearly, as contiguous vertices always become adjacent, adding a contiguous vertex to a component can never divide it up into two components.

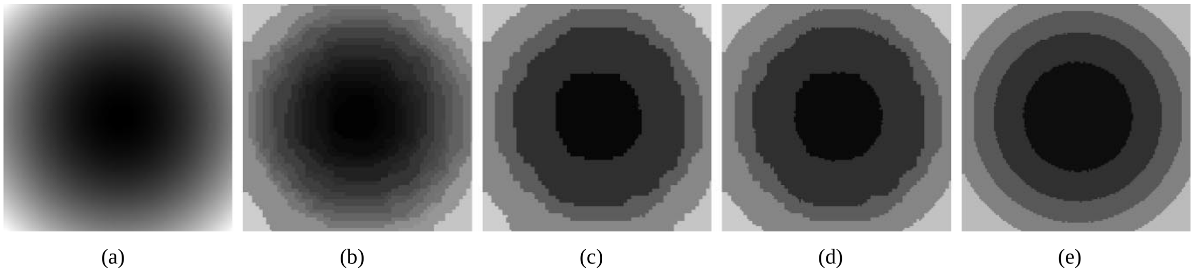


Fig. 7. CCA applied to the Circle image (a) with $\sigma^2_{\max} = 20^2$. (b)–(e) Results after 5, 10, 50, 1000 epochs, respectively.

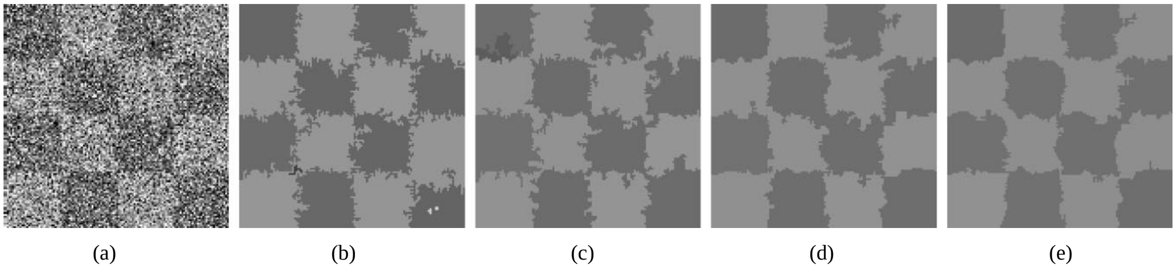


Fig. 8. Results of applying the CCA segmentation method to the Checkerboard image with various filter size $f$ and corresponding $\sigma^2_{\max}$ settings. (a) Original. (b) No filter, $\sigma^2_{\max} = 60^2$, 1067 epochs. (c) $f = 3$, $\sigma^2_{\max} = 20^2$, 1357 epochs. (d) $f = 5$, $\sigma^2_{\max} = 14^2$, 784 epochs. (e) $f = 7$, $\sigma^2_{\max} = 10^2$, 696 epochs.

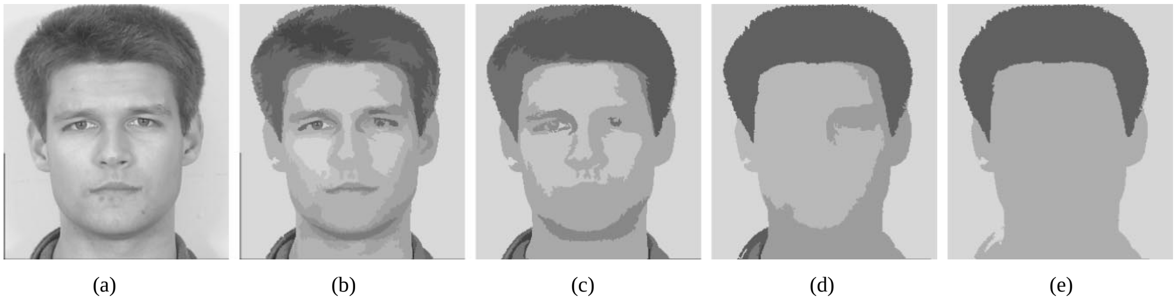


Fig. 9. CCA segmentation results with different scale parameter settings. (a) Original. (b) $\sigma^2_{\max} = 10^2, M = 72$. (c) $\sigma^2_{\max} = 15^2, M = 7$. (d) $\sigma^2_{\max} = 20^2, M = 7$. (e) $\sigma^2_{\max} = 25^2, M = 3$

are used to settle the segment borders. As expected, the final result in Fig. 7(e) shows a number of clear ring-shaped segments.

Next, we did an experiment to illustrate how to handle noise properly with the CCA segmentation method. We used a 128 × 128 Checkerboard image with squares having either pixel intensity 100 or 160. We added Gaussian noise with $\mu = 0$ and $\sigma^2 = 60^2$ to this image; see Fig. 8(a). Fig. 8(b)–(e) show a number of segmentation results with different pre-filter settings. Since the noise level ($\sigma^2 = 60^2$) and the size of the pre-filter ($f$ in Fig. 8) are known, $\sigma^2_{\max}$ could be adjusted accordingly using (8). Note that the noise component has been divided by $f^2$, since a filtering in both the horizontal and vertical direction took place. Additionally, $\sigma^2_{\max}$ was set slightly higher than prescribed in (8), because the variance in the homogeneous regions increased due to blurring effects of the filter. The figures clearly show that the pre-filtering results in smoother segment borders.

When we described our model design considerations, we stated that especially in image segmentation it is not desirable to have the number of segments as model parameter. Instead, a scale parameter was introduced to select a maximum allowed variability per segment. In Fig. 9, we show the results of varying the scale when applying the CCA segmentation method to a face image. Clearly, when the value of $\sigma^2_{\max}$ is increased the smaller details disappear from the resulting segmented images. In Fig. 9(b) most facial features are still visible, since their segment variance is below the segment variance constraint. In Fig. 9(c) and (d) the details gradually disappear, while at the scale in Fig. 9(e) only the background, face, and hair can be distinguished.

### B. Comparison With Other Methods

In the next experiments, we compared the proposed method to two other typical segmentation methods. The first one is the “split-and-merge with grouping” algorithm, which has a similar model and parameters. The second one is a Markov random field model-based segmentation algorithm. The latter segmentation method has quite a different model and is often used for segmentation and restoration. Since obtaining the ground truth is difficult in image segmentation, comparing segmentation results is not trivial. Here, we compare the results with respect to the following qualitative criteria: 1) *accurateness*: the segment border represents a true contour. 2) *continuity*: the segment border may not contain holes if the object that it represents has a continuous contour. 3) *fragmentation*: the union of neighboring segments must be inhomogeneous.

Before discussing the comparative experiments, we first describe the other two segmentation algorithms in more detail.

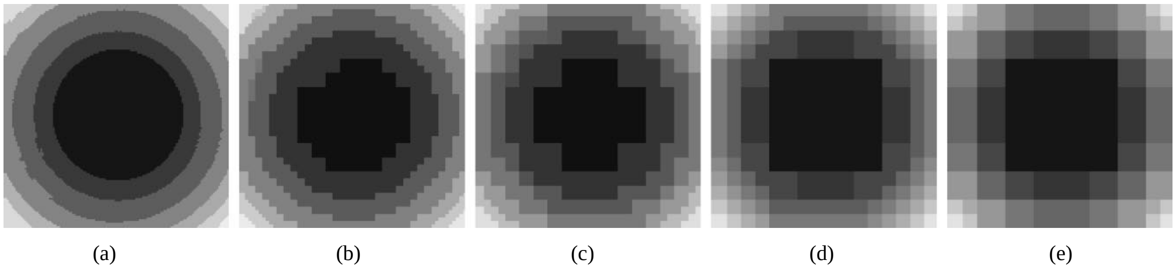


Fig. 10. Results of applying the SMG algorithm to the Circle image [Fig. 7(a)] with various $\sigma^2_{\max}$ and $\delta_\mu$ parameter settings. (a) $\sigma^2_{\max} = 0, \delta_\mu = 22$. (b) $\sigma^2_{\max} = 5^2, \delta_\mu = 20$. (c) $\sigma^2_{\max} = 10^2, \delta_\mu = 20$. (d) $\sigma^2_{\max} = 15^2, \delta_\mu = 15$. (e) $\sigma^2_{\max} = 20^2, \delta_\mu = 0$.

### *Split-and-Merge With Grouping Segmentation (SMG)*

The split-and-merge implementation we use is based on [21] and [31]. Typically the split-and-merge algorithm uses a scheme in which a square region is recursively split into four quads if the homogeneity predicate is false. Merging is allowed if for neighboring regions the homogeneity predicate is true. The algorithm stops when no splitting or merging is possible. The homogeneity predicate we use is the Chi-squared test, where the rejection probability is set to 0.5. The maximum variance $\sigma^2_{\max}$ used in this test has similar semantics as the maximum variance in (4). The specific implementation uses a quad tree structure for the splitting and merging. Since only merges on the same level of the quad tree are allowed, a final grouping step is incorporated. In the grouping stage all regions that together are homogeneous are grouped together. Now, regions are homogeneous if the distance between their means is below a certain maximum $\delta_\mu$. Finally, the method usually ends up with many very small regions at the intended segments borders. These regions are grouped together with their neighboring segments if their size is relatively small (default 0.2%) with respect to their largest neighbor. Like the proposed method in this paper, the split-and-merge algorithm does not need to know the number of segments *a priori*. Moreover, segment connectivity is always satisfied, i.e., the computed segment statistics are always based on connected components. The scale and noise can be regulated by setting $\sigma^2_{\max}$ and $\delta_\mu$ properly. Clearly, setting $\sigma^2_{\max}$ very low yields many small regions, so the actual segmentation is done in the grouping stage. In this way the algorithm turns into a region-growing algorithm. The consequences of variations in both parameters will become clear in the experiments.

### *Markov Random Field Model-Based Segmentation (MRF)*

In Markov random field models for image segmentation, typically an energy function with two terms is minimized simultaneously for all pixels in the image. It is assumed that the number of color distributions is known together with their mean $\mu_i$ and variance $\sigma^2_i$. For the natural images, we estimate these values with a mixture of Gaussians method, though we fix $\sigma^2_i = \sigma^2_{\max}$ to make a more fair comparison with the other methods. For the synthetic images we feed the algorithm the true color distribution values. Then, one of the terms in the energy function accounts for the distance between the pixel color and the color distribution to the pixel color and the other term introduces contextual information through the Markov random field (MRF). That is, the second term imposes the same segment label on pixels in a certain neighborhood (in this case a second-order model). The neighborhood term is weighted with a factor $\beta$ to stress the relative importance of local connectivity and color matching. Since global optimization of the model is intractable, several approximations have been proposed [5], [10], [17]. We use the well-known ICM method, due to [5], which has been proven to be the most effective method to optimize the MRF-based model [13]. As remarked earlier, as a consequence of the optimization of these pixel criteria, this is rather a quantization than a segmentation method. The scale and noise adaptation is controlled both by the order of the Markov random field (second order) and the number of pre-defined color distributions.

### *Gradual Changing Color*

First, we applied the SMG and MRF methods to the synthetic images that we already used to explore the CCA method in the previous subsection. For the original SMG method the Circle image is especially difficult because the intensity changes gradually. Only if the $\sigma^2_{\max}$ parameter is set to zero the result is similar to the CCA outcome [Fig. 10(a)]. When, on the other hand, $\sigma^2_{\max}$ is set “properly,” the resulting segments become irregular, as can be seen in Fig. 10(b)–(e). The MRF method has the best results when the contextual information is made unimportant ($\beta = 0.1$) as in Fig. 11(a). Then, the segmentation result is mainly based on the image intensities. In this case, where there is no noise, there is strong dependence between image intensity and segment label. Accordingly, the contextual information is indeed superfluous or even harmful; see Fig. 11(b)–(e).

### *Noisy Image*

Next, we applied the algorithms to the noisy Checkerboard image in Fig. 8(a). The SMG algorithm clearly has problems with this image. We varied the $\sigma^2_{\max}$ parameter between $\sigma^2_{\max} = 0$ and the true variance of the Gaussian noise ($\sigma^2_{\max} = 60^2$), and adjusted $\delta_\mu$ as to achieve the best possible segmentation result. The results achieved with $\sigma^2_{\max} = 60^2$ fit quite well to the original noiseless image [Fig. 12(e)]. We must, however, note that the regular block pattern of the image definitely favors the split-and-merge algorithm. With lower $\sigma^2_{\max}$ values irregular borders occur and, as a consequence of the greedy chaining of small regions, many extra irregular segments occur; see Fig. 12(a)–(e). Also the MRF method is able to find useful segmentation results for the Checkerboard image. However, when the $\beta$ parameter is (too) low, the segments are strongly fragmented (connectivity violated) and when $\beta$ is high, blurring or smearing effects occur; see Fig. 13(a)–(e).

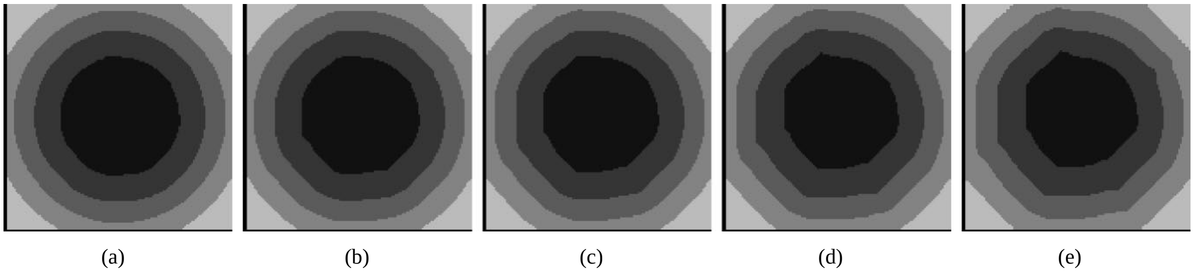


Fig. 11. Results of applying the MRF algorithm to the Circle image [Fig. 7(a)] with various $\beta$ parameter settings. (a) $\beta = 0.1$. (b) $\beta = 0.2$. (c) $\beta = 0.3$. (d) $\beta = 0.4$. (e) $\beta = 0.5$.

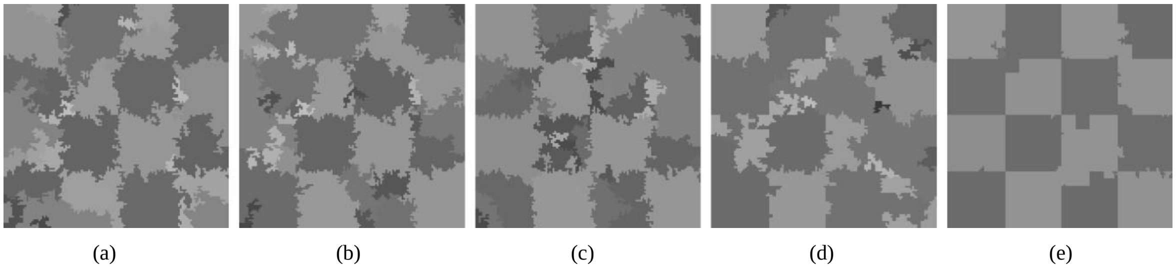


Fig. 12. Results of applying the split-and-merge segmentation method to the Checkerboard image with various $\sigma^2_{\max}$ and $\delta_\mu$ settings. (a) $\sigma^2_{\max} = 0, \delta_\mu = 55$. (b) $\sigma^2_{\max} = 10^2, \delta_\mu = 55$. (c) $\sigma^2_{\max} = 20^2, \delta_\mu = 55$. (d) $\sigma^2_{\max} = 50^2, \delta_\mu = 45$. (e) $\sigma^2_{\max} = 60^2, \delta_\mu = 25$.

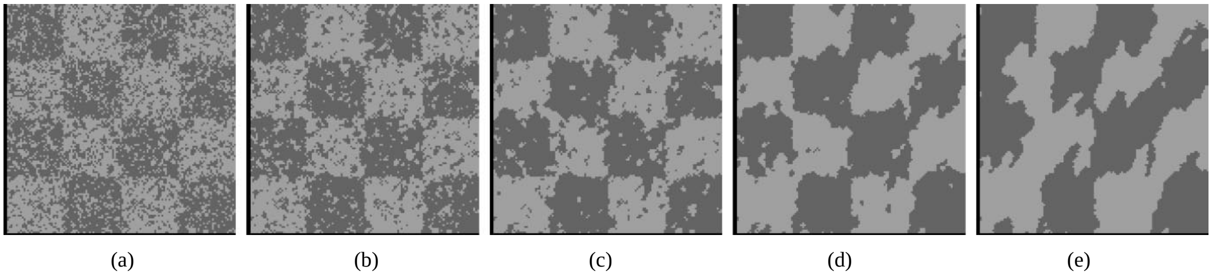


Fig. 13. Results of applying the MRF model-based segmentation method to the Checkerboard image with various $\beta$ settings and $\mu_1 = 100, \mu_2 = 160$, and $\sigma_1^2 = \sigma_2^2 = 60^2$. (a) $\beta = 0.1$. (b) $\beta = 0.2$. (c) $\beta = 0.3$. (d) $\beta = 0.4$. (e) $\beta = 0.5$.

### *Natural Images*

Finally, we applied all three algorithms to a number of natural images. In the natural images a number of difficult aspects of image segmentation come together, like noise, gradual changing colors, and large and small objects. For all algorithms we chose those parameter settings that turned out to be the best for each of them. The $\beta$ parameter of the MRF algorithm was fixed to $\beta = 0.3$. We used $\sigma^2_{\max}$ of the SMG equal to $\sigma^2_{\max}$ of the CCA to select a certain scale. Additionally, we set $\sigma^2_{\max} = 0$ to turn the SMG into a region-growing algorithm. In all cases the $\delta_\mu$ parameter of the SMG algorithm is adjusted accordingly.

We started with the Orca image in which the (splashing) water and the sky contain gradually changing colors; see Fig. 14(a). Further, there are some well-defined large and small segments on the Orca. Like in the Circle image the MRF algorithm creates artefacts in the gradually changing colors, as can be seen in Fig. 14(b). Besides, it creates many very small regions around the Orca, which is, however, hard to see in the figure. When $\sigma^2_{\max}$ is set to the proper scale value, the SMG algorithm has contour artefacts; see Fig. 14(c). The SMG with $\sigma^2_{\max} = 0$ makes nice contours, but has some artefacts resulting from the chaining of small regions, like the light-gray areas on the trunk of the Orca and the light-gray region above its fin; see Fig. 14(d). The CCA has a good segmentation result, except that (as for the SMG method) some details seem missing, like the small white area on top of the Orca's head. When we measured the corresponding areas it appeared that the variance was indeed below the given $\sigma^2_{\max}$, hence, the details should indeed not be visible at this scale.

The image from the Flower-Garden sequence (Fig. 2) has both very large segments (the tree) as well as many small segments (the flowers). The MRF model-based method shows the expected smearing effects; see Fig. 15(a). We had to lower the small-region-removal parameter of the SMG method to 0.01% in order to prevent all flowers from being removed. Besides, with $\sigma^2_{\max} = 15^2$ the same blocking artefacts occurred as before; see Fig. 15(b). Configured as a region grower ($\sigma^2_{\max} = 0$) the SMG results are better [Fig. 15(c)], though in all cases small noisy regions remain. These are not always easy to see in the figures. The CCA method gives segmentation results with both small details and large homogeneous regions.

Finally, we applied the algorithms to a noisy image [Fig. 16(a)]. The results are similar to those obtained with the Checkerboard image. That is, the MRF has good segmentation results apart from the smearing effects [Fig. 16(b)] and the SMG has again blocking artefacts [Fig. 16(c) and (d)]. As expected, the SMG with $\sigma^2_{\max} = 0$ [Fig. 16(d)] and the CCA [Fig. 16(e)] give similar irregular contours.

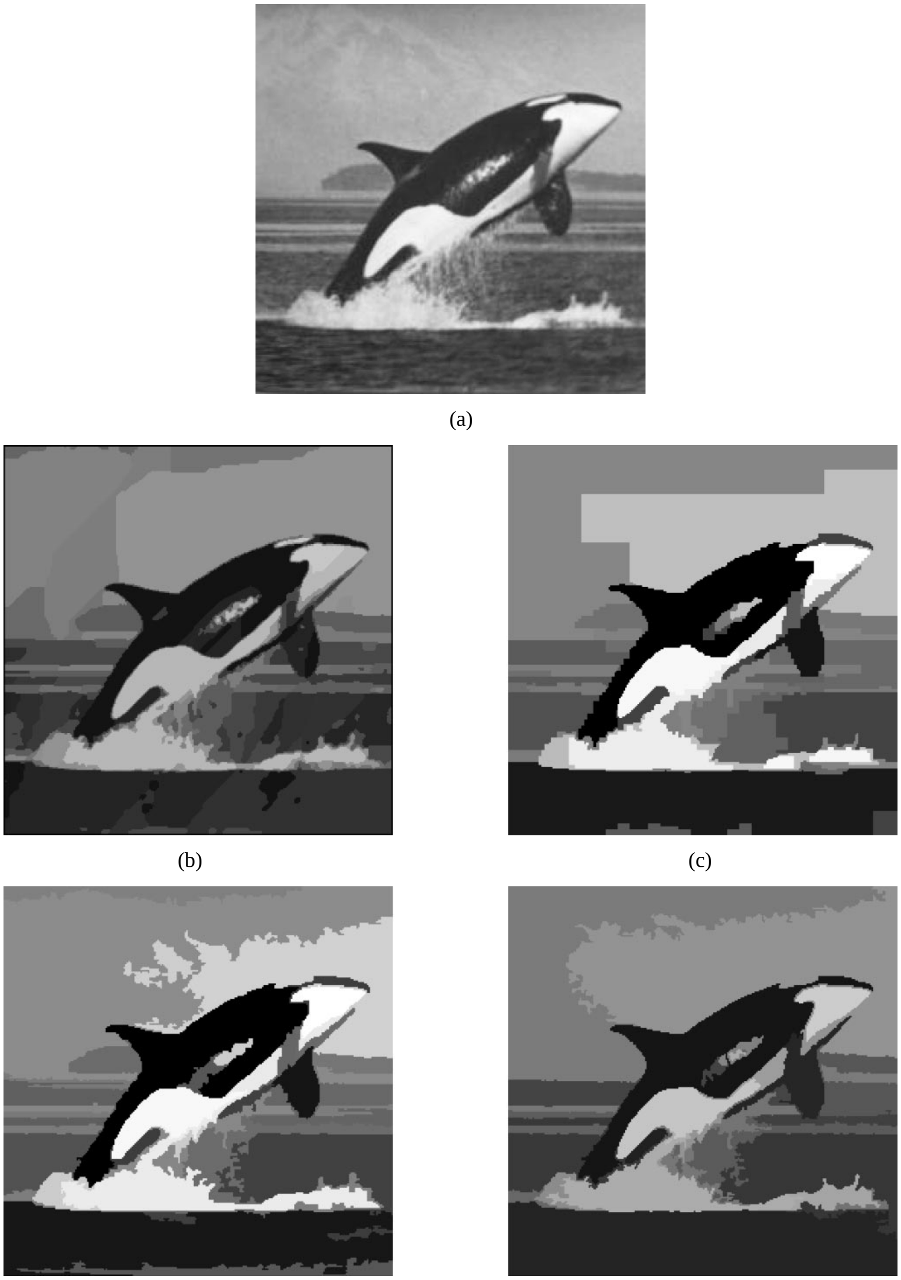
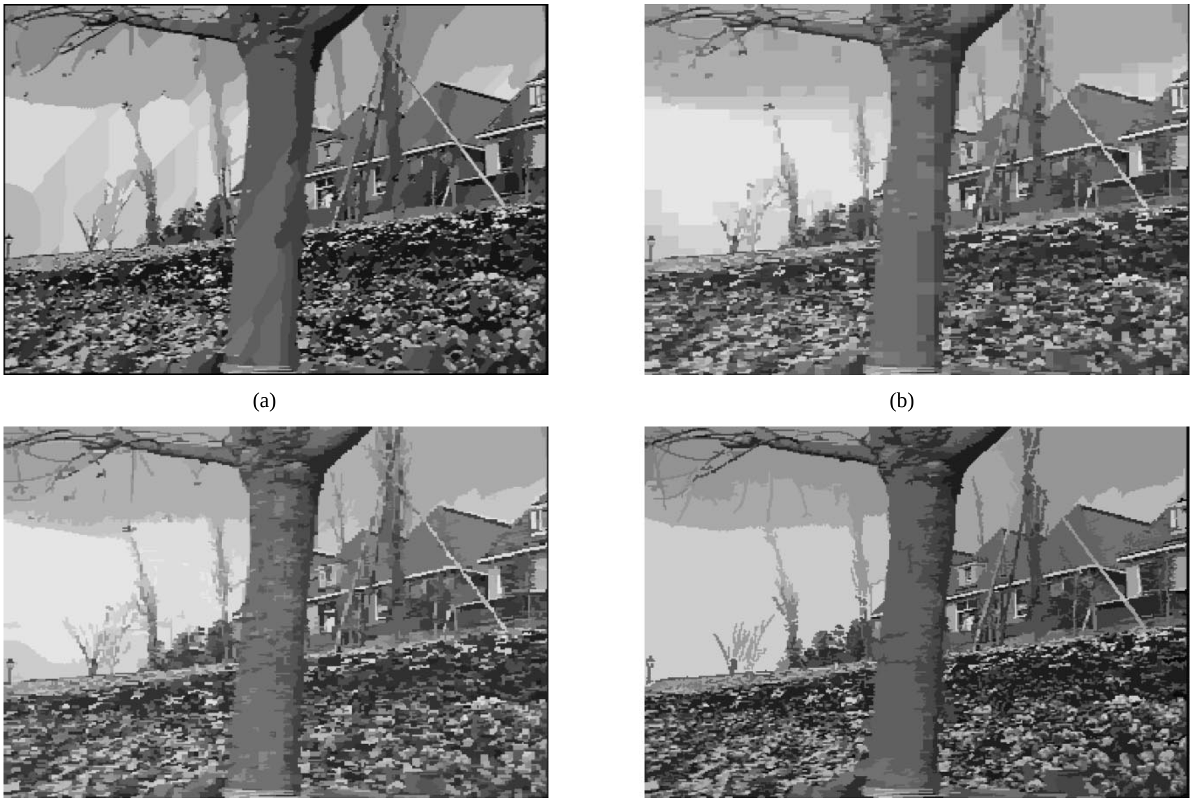


Fig. 14. (a) Shows the Orca image. In (b)–(e) the segmentation results are shown with different algorithms and parameter settings. (a) Original. (b) MRF $\sigma^2_{\max} = 15^2$, $\beta = 0.3$. (c) SMG $\sigma^2_{\max} = 15^2$, $\delta_\mu = 5$. (d) SMG $\sigma^2_{\max} = 0$, $\delta_\mu = 15$. (e) CCA $\sigma^2_{\max} = 15^2$.

## VI. Conclusions

In this paper we presented a cluster model for which the number of clusters does not have to be known *a priori*. Instead of the number of clusters, we introduced a scale parameter. We justified this choice among others from the image segmentation domain, which we especially addressed. We specialized the cluster model for the image segmentation problem and presented a *cellular coevolutionary algorithm* (CCA) to optimize it in a distributed way. In the experiments, we showed the effectiveness of the method and compared it to two other well-known segmentation methods. Both the Markov random field model based segmentation method (MRF) and the split-and-merge-and-group (SMG) scheme have quite a number of parameters, some of which we estimated and fixed and others which we varied in the experiments. Though it is hard to draw conclusions without trying all combinations of parameter settings, the proposed CCA is clearly simpler since it has only one essential parameter. The advantage of such a simple model is that it is both easier to tune and to optimize. Having said this, we conclude the following. The MRF method implements segment connectivity as a *soft constraint* and uses a weighting factor $\beta$ to stress the importance of connectivity. Especially in case of noise and gradual segment transitions, connectivity must be stressed since the pixels are not linked based on intensity only. However, the experiments show that setting $\beta$ too high results in smearing and blurring artefacts and setting it too low results in fragmented segments, though it has to be noted that the MRF artefacts are not only caused



(a) (b) (c) (d)

Fig. 15. (a)–(d) Show segmented images resulting from applying different algorithms with several parameter settings to an image from the Flower-Garden sequence. Fig. 2 shows the original image. (a) MRF $\sigma^2_{\max} = 15^2$, $\beta = 0.3$. (b) SMG $\sigma^2_{\max} = 15^2$, $\delta_\mu = 20$. (c) SMG $\sigma^2_{\max} = 0$, $\delta_\mu = 15$. (d) CCA $\sigma^2_{\max} = 15^2$.

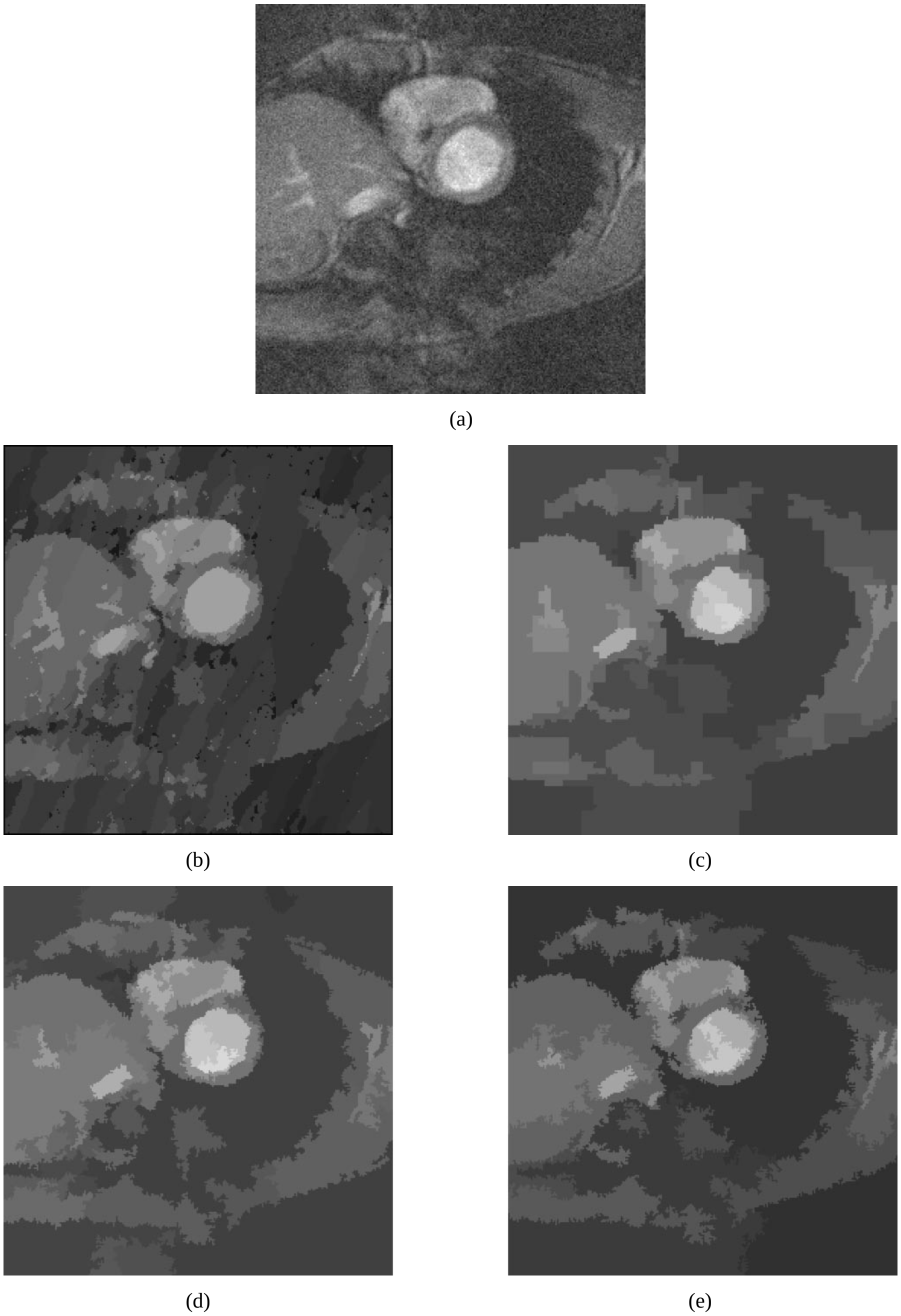
by the MRF model, but also by the (ICM) model optimization scheme. The right trade-off between connectivity and correct border estimation can be hard to find. Though connectivity is a hard constraint for the SMG method, this method also has significant artefacts. First, it strongly needs a small-region elimination step. This is a problem if the image indeed contains small and large segments. Further, if the $\sigma^2_{\max}$ value, which serves to define segment homogeneity, is set properly, then strong blocking effects occur. On the other hand, when it is set very low the SMG method is actually a region-growing algorithm. Then, the results are usually quite good, though borders can be irregular and the grouping procedure has order dependence and chaining artefacts. As regards the SMG method with low $\sigma^2_{\max}$ values, the CCA tends to result in somewhat irregular contours when the image is noisy. However, if an estimate of the amount of noise is known, the proposed method prescribes to pre-filter the image and to adjust $\sigma^2_{\max}$ accordingly, which indeed results in smoother contours. Another problem that the CCA method suffers from is that small regions can be absorbed by large homogeneous regions. This is a consequence of the variance-based scale constraint in the proposed model. The SMG method lacks the same problem in the merging and grouping steps. Other criteria can be developed to prevent this, though it remains a difficult issue.

There are a number of possible extensions to the proposed method. First, the efficiency of the CCA needs improvement, especially if image sequences are to be segmented. The algorithm uses multiple agents that locally optimize a criterion. Accordingly, parallel computer architectures can be exploited very effectively. In image sequences the efficiency can be further improved by starting the evolution with the resulting agent constellations from the previous image. Additionally, it would be very interesting to estimate a texture model for each segment. This is far from trivial since the determination of textured segments conflicts with finding small segments. Another appealing idea is to develop a similar algorithm for the general clustering problem. The main difficulty in that respect is to exploit locality, which is especially easy in the image domain.

## Appendix
## Cut Vertex Detection Algorithm

Detecting whether a segment component becomes disconnected after the deletion of a vertex $\boldsymbol{x}$ implies detecting whether $\boldsymbol{x}$ is a cut vertex of a segment component $S$. The vertex degree of $\boldsymbol{x} \in V(S)$ is at most three, because otherwise it could not be contiguous to another component. If the vertex degree $d(\boldsymbol{x}) = 1$, $\boldsymbol{x}$ cannot be a cut vertex, we are done. For the other cases, we only need to know if there is a path between every pair of adjacent vertices of $\boldsymbol{x}$ which does not include $\boldsymbol{x}$ itself. If there is no such path, the graph will be disconnected after the deletion of $\boldsymbol{x}$, i.e., it is a cut vertex.

First set the visit state of all vertices in $V(S)$ to $state = 0$. Search depth first, starting at an arbitrary adjacent vertex, say $\boldsymbol{a}_1$, of $\boldsymbol{x}$ and visit all adjacent vertices of every visited vertex. The searching stops when another (arbitrary) adjacent vertex of


(a)

(b)

(c)

(d)

(e)

Fig. 16. (a) Shows an MRI image with extra artificial noise. In (b)–(e) the segmentation results are shown with different algorithms and parameter settings. (a) Original. (b) MRF $\sigma^2_{\max} = 15^2$, $\beta = 0.3$. (c) SMG $\sigma^2_{\max} = 15^2$, $\delta_\mu = 10$. (d) SMG $\sigma^2_{\max} = 0$, $\delta_\mu = 15$. (e) CCA $\sigma^2_{\max} = 15^2$.

$\boldsymbol{x}$, say $\boldsymbol{a}_2$, is found or all connected vertices to $\boldsymbol{a}_1$ are visited. While searching, set the visit state of vertices to $state = 1$ and skip those vertices that have already been visited ($state = 1$). If no path from $\boldsymbol{a}_1$ to an adjacent vertex of $\boldsymbol{x}$ is found, then $\boldsymbol{x}$ is a cut vertex.

Otherwise, if there is one more adjacent vertex, say $\boldsymbol{a}_3$, start searching from this vertex and continue until a vertex is found with $state = 1$ or until all connected vertices to $\boldsymbol{a}_3$ are visited. Again mark evaluated vertices while searching, but now set $state = 2$ and skip those vertices that already have $state = 2$. Only if there was no path to a vertex with $state = 1$ then $\boldsymbol{x}$ is a cut vertex.

Additionally, we speed up the algorithm considerably by using depth-first search with iterative deepening, because usually the path connecting adjacent vertices is very short.

## References

[1] P. Andrey and P. Tarroux, “Unsupervised image segmentation using a distributed genetic algorithm,” *Pattern Recognit.*, vol. 27, no. 5, pp. 659–673, 1994.

[2] ——, “Unsupervised segmentation of Markov random field modeled textured images using selectionist relaxation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 252–262, Mar. 1998.

[3] A. Baraldi and P. Blonda, “A survey of fuzzy clustering algorithms for pattern recognition—Part I,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, no. 6, pp. 778–785, 1999.

[4] ——, “A survey of fuzzy clustering algorithms for pattern recognition—Part II,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, no. 6, pp. 786–801, 1999.

[5] J. Besag, “On the statistical analysis of dirty pictures,” *J. R. Statist. Soc. B*, vol. 48, pp. 259–302, 1986.

[6] J. C. Bezdek and N. R. Pal, “Some new indexes of cluster validity,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 3, pp. 301–315, 1998.

[7] S. M. Bhandarkar and H. Zhang, “Image segmentation using evolutionary computation,” *IEEE Trans. Evol. Comput.*, vol. 3, pp. 1–21, Apr. 1999.

[8] B. Bhanu, S. Lee, and S. Das, “Adaptive image segmentation using genetic and hybrid search methods,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, pp. 1268–1291, Oct. 1995.

[9] D. E. Brown and C. L. Huntley, “A practical application of simulated annealing to clustering,” *Pattern Recognit.*, vol. 25, no. 4, pp. 401–412, 1992.

[10] P. B. Chou and C. M. Brown, “The theory and practice of Bayesian image labeling,” *International Journal of Computer Vision*, vol. 4, pp. 185–210, 1990.

[11] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1, pp. 224–227, Apr. 1979.

[12] A. di Nola, V. Loia, and A. Staiano, “Genetic-based spatial clustering,” in *Proc. 9th IEEE Int. Conf. Fuzzy Systems*, 2000, pp. 953–956.

[13] R. C. Dubes, A. K. Jain, S. G. Nadabar, and C. C. Chen, “MRF model-based algorithms for image segmentation,” in *Proc. IEEE 10th Int. Conf. Pattern Recognition*, vol. I, 1990, pp. 808–814.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.

[15] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” *J. Cybern.*, vol. 3, pp. 32–57, 1974.

[16] ——, “Well separated clusters and optimal fuzzy partitions,” *J. Cybern.*, vol. 4, pp. 95–104, 1974.

[17] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721–741, 1984.

[18] L. O. Hall, I. B. Özyurt, and J. C. Bezdek, “Clustering with a genetically optimized approach,” *IEEE Trans. Evol. Comput.*, vol. 3, pp. 103–112, July 1999.

[19] M. Haseyama, M. Kumagai, and H. Kitajima, “A genetic algorithm based image segmentation for image analysis,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 6, 1999, pp. 3445–3448.

[20] Y. Horita, T. Murai, and M. Miyahara, “Region segmentation using K-mean clustering and genetic algorithms,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 1994, pp. 1016–1020.

[21] S. L. Horowitz and T. Pavlidis, “Picture segmentation by a tree traversal algorithm,” *J. ACM*, vol. 23, no. 2, pp. 368–388, 1976.

[22] L. J. Hubert and P. Arabie, “Comparing partitions,” *J. Class.*, vol. 2, pp. 193–218, 1985.

[23] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[24] R. W. Klein and R. C. Dubes, “Experiments in projection and clustering by simulated annealing,” *Pattern Recognit.*, vol. 22, no. 2, pp. 213–220, 1989.

[25] K. Krishna and M. N. Murty, “Genetic K-means algorithm,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 433–439, June 1999.

[26] G. N. Lance and W. T. Williams, “A general theory of classificatory sorting strategies: II clustering systems,” *Comput. J.*, vol. 10, pp. 271–277, 1967.

[27] T. Van Le, “Evolutionary fuzzy clustering,” in *Proceedings IEEE International Conference on Evolutionary Computation*. Killington, VT: IEEE Computer Society Press, Oct. 14–16, 1995, vol. 2, pp. 753–758.

[28] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, vol. 1, L. M. Le Cam and J. Neyman, Eds., 1967, pp. 281–297.

[29] T. N. Pappas, “An adaptive clustering algorithm for image segmentation,” *IEEE Trans. Signal Processing*, vol. 40, pp. 901–914, Apr. 1992.

[30] E. J. Pauwels and G. Frederix, “Finding salient regions in images: Nonparametric clustering for image segmentation and grouping,” *Comput. Vis. Image Understand.*, vol. 75, no. 1, 2, pp. 73–85, 1999.

[31] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer-Verlag, 1977.

[32] S. Z. Selim and K. Alsultan, “A simulated annealing algorithm for the clustering problem,” *Pattern Recognit.*, vol. 24, no. 10, pp. 1003–1008, 1991.

[33] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. New York: Academic, 1999.

[34] R. Wilson and G. H. Granlund, “The uncertainty principle in image processing,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 758–767, Nov. 1984.

[35] M. Yu, N. Eua-anant, A. Saudagar, and L. Udpa, “Genetic algorithm approach to image segmentation using morphological operations,” in *Proc. IEEE Int. Conf. Image Processing*, 1998, pp. 775–779.

[36] L. Zhao, Y. Tsujimura, and M. Gen, “Genetic algorithm for fuzzy clustering,” in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1996, pp. 716–719.

**Cor J. Veenman** received the B.Eng. degree in electrical engineering from the HTS, Hilversum, The Netherlands, in 1987 and the M.Sc. degree in computer science from the Free University, Amsterdam, The Netherlands, in 1996. In 2002, he received the Ph.D. degree for his work in electrical engineering from Delft University of Technology, Delft, The Netherlands.

From 1991 to 1997, he was a Software Developer in multimedia applications and taught programming courses. In 1997, he started as a Research Assistant at the Delft University of Technology. His research interests are in optimization and artificial intelligence and in applying related techniques in pattern recognition, computer vision, and image segmentation.

**Marcel J. T. Reinders** received the M.Sc. degree in applied physics and the Ph.D. degree in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 1990 and 1995, respectively.

Currently, he is an Associate Professor at Delft University of Technology. His research interests are in computer vision and machine learning. Recently, he became interested in the rapidly growing field of bioinformatics that uses machine learning techniques to study biological phenomena, such as the behavior of gene expressions or protein mapping.

**Eric Backer** (M’82) received the M.Sc. and Ph.D. degrees from the Delft University of Technology, Delft, The Netherlands, in 1969 and 1978, respectively.

He was appointed a Full Professor at Delft University of Technology in 1982, and was a Visiting Professor at Michigan State University, Ann Arbor, in 1978 and 1982, and at the University of South Florida in 1993. His current research interests include pattern recognition, image processing, and machine intelligence.

Dr. Backer is the Managing Editor of *Pattern Recognition Letters* and has been the General Cochairman of the 11th ICPR.