3. Mathematical morphology of binary images
CHAPTER 3

MATHEMATICAL MORPHOLOGY OF BINARY IMAGES

"By synthesizing the mechanisms underlying the evolutionary process in computers and other 'non biological' media, we can discover solutions to engineering problems that have long resisted our traditional approaches"
- Langton C.G.

3. 1. Overview

Vision is ruled by principles, such as perceptual grouping, selection and discriminations, which mostly depend on regularities of nature such as cohesiveness of matter or existence of bounding surfaces [Marr, 1982]. As these properties are valid also for structures present in images, they can be exploited to build segmentation systems for such images. If no other source of information is used, the resulting segmentation algorithms are independent of acquisition parameters. In the last few years, a variety of evolutionary approaches have been applied to the problem of discovering algorithms for image processing, so much so that evolutionary image processing can almost be considered as a separate field of research. Mathematical Morphology (MM) is well known as a powerful tool for various image processing tasks [Harvey and Marshall, 1996]. It is suitable for shape related processing since morphological operations are directly related to the object shape.

MM [Serra, 1982] is a reasonably vital part of image analysis. The non-morphological approach to image processing is close to calculus, being largely based on the point spread function concept. (e.g. Dirac impulse) and linear transformations such as convolution. MM is based on geometry and shape; morphological operations
simplify images, and preserve the main shape characteristics of objects. MM is basically a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries and skeletons. Morphological techniques are also used for pre- or post-processing, such as morphological filtering, filtering, thinning and pruning.

To design a MM procedure (i.e. algorithm) some expert knowledge is necessary to properly select the structuring elements and to make an adequate selection of the morphological operators sequence. It has been suggested that Genetic Algorithms [Crimmins and Brown, 1985] could have good performance for morphological filter design. Furthermore, GAs have obtained optimistic results in the automatic acquisition of MM procedures [Yoda et al., 1999]. Previous research had been limited to optimizing one filter (for gray-level images) [Yu et al., 1998] and to obtain fixed-length sequences of morphological operators using a reduced number of structuring elements [Bala and Welchsler, 1991].

Morphological operations are predominantly used for image pre-processing (noise filtering, shape simplification), enhancing object structure (skeletonizing, thinning, thickening, convex hull, object and making) and quantitative description of objects (area, perimeter, projections, Euler-Poincare characteristics) [Dougherty and Latufo, 2003]. In an attempt to automate morphological filter design for target detection in grey-scale image, Nong et al., [2000] used chromosomes with a fixed number of components, and the pixel-wise square error as their fitness function, proposing also new crossover and mutation operators to improve convergence.
3. 2. Basic Morphological Operations

Our approach used the operators used in GENIE, an evolutionary system for finding features of interest in multi-spectral remotely-sensed images. [Brumby et al., 1999]. In this system a set of morphological operators was provided as part of a larger set of image processing primitives. The sizes and shapes of the structuring elements used by these operators were restricted to a pre-determined set of classical primitive shapes. So this work use a small set of regular kernels. The two morphological operations most used in MM are dilation and erosion and most of the algorithms developed by experts to perform a particular task make use of them.

With A and B as sets in \( Z^2 \) and \( \phi \) denoting the empty set, the dilation of A by B, denoted by \( A \oplus B \), consists of obtaining the reflection of B about its origin, then shifting this reflection \( B_1 \) by \( x \) to obtain \((\hat{B})_x\). The dilation of A by B, as shown in Equation 3.1 is the set of all \( x \) displacements such that \((\hat{B})_x\) and A overlap by at least one nonzero element. Figure 3.1 exemplifies this operation.

\[
A \oplus B = \{x / (\hat{B})_x \cap A \neq \phi\} \quad (3.1)
\]

\[
A \odot B = \{x / (\hat{B})_x \subseteq A\} \quad (3.2)
\]

For sets A and B in \( Z^2 \) the erosion of A by B, denoted \( A \odot B \), is defined as the set of all points \( x \) such that \( \hat{B} \) translated by \( x \) is contained in A. The erosion of A by B, as shown in Equation 3.2 is exemplified in Figure 3.2. These two equations are not the only definitions for dilation and erosion, but they are usually preferred in practical implementations because of their analogy with the operation of convolution for linear filtering.

43
Figure 3.1 Sub-image representation of Dilation

Figure 3.2 Sub-image representation of Erosion
3.3. Procedural Design

With a wide range of applications, morphological filters are an important class of non-linear digital signal processing and analysis filters giving excellent results in areas such as noise reduction, edge detection and object recognition. However, few design methods exist for these morphological filters, with selection of filter type and sequence of application tending to be both adhoc and application specific. The number of possible structuring elements (also known as kernels) from which to choose is very large and this, combined with the possible choices of morphological operator sequences, indicates the complexity of the morphological filter design.

In the case of morphological filters there are two basic parameters:

i. the structuring elements and
ii. the sequence of morphological operations.

It has been demonstrated how simple GAs can be employed in the search for morphological filters realizing optimum performance in a given image processing task [Harvey and Marshall, 1996]. In their approach a fixed number of morphological operators including a do-nothing operator was used to avoid the problem of using only fixed sized sequences. It seems that this do-nothing operator appear very often in the chromosomes, indicating how important it is to let the search use fixed-length parameters.

It has been considered to obtain MM procedures for an original binary image and a hand-made goal image. The search space of morphological sequences is explored using GAs. The parameter representation uses 4-20 fixed-length
chromosomes with four types of regular structuring elements. After 4 examples of different complexity, it is clear that the proposed method is fairly good to obtain a near optimum solution, but the required computational effort increases when the task to perform is more complex. Fig.3.3 shows how it is expected to get a goal image from the original after a particular transformation procedure. In the following section we show how GA can be used to overcome the problems presented.

![Original Image(O) vs Goal Image(G)](image)

Fig. 3.3 Transformation from Original Image (O) to Goal Image(G)

The GA approach suggested assumes that it is possible to find a sequence of morphological operators in the MM algorithm’s search space to convert an image into another containing only a particular feature of interest. To show this idea we select musical sheets as examples and extract some features from them. The process is visualized in Fig. 3.4.
Algorithm 3.1 for evolving the best sequence of parameters

1. Create some examples of correct feature extraction by hand to be used as training sets in GA.

2. Define the type, size and number of structuring elements to be used in the GA search.

3. Define the primitive parameter set

4. Start the GA search to obtain a (near) optimum tree representing a MM algorithm sequence.

5. Select the best evolved sequence, according to the fitness value assigned for the fitness function.

6. Analyze the feature visually to decide whether or not the numerical fitness value is reflected in the image quality.

The features used in this process are described in the next subsections.
3.3.1. Structuring Elements

When a MM algorithm is designed by hand, we usually chooses a regular structuring element to make a sequence of operations, there is no reason for choosing a regular elements except that we as humans, are more likely to understand regularities such as squares, lines, triangles, etc. It is interesting to note that the MM algorithm’s search space is not limited to regular structuring elements, but that there are many irregular structuring elements usually ignored when designing a MM algorithm. We suggest to include irregular structuring elements (selected randomly) in the GA search space. We use regular structuring elements of sizes 3 x 3 and 5 x 5 such as those shown in Fig. 3.5.

![Regular structuring elements of size 3 x 3 and 5 x 5](image)

3.3.2. Training Sets

We test GA to search for MM algorithms looking for procedures to extract three different features: heads and hooks. To find out whether or not the size of the images in the training sets affects the results quality, we made by hand training sets
containing images of size 16 x 16 and 32 x 32 each one of the different features to look for. We use images belonging to the 16 x 16 training set. To learn if the number of elements in the training set affects the results quality we use two different numbers of elements in the training set (1 and 5).

3.3.3. Fitness Functions
All our experiments were performed using two different fitness functions to evaluate the similarity between two images. The objective fitness function A is known as similarity \((0 \leq A \leq 1)\) [Yoda et al., 1999], a normalized correlation coefficient between a goal and a processed image defined as

\[
A = \frac{(f.g)}{\sqrt{(f.f)\sqrt{(g.g)}}}; \text{where} (f.g) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} f(i,j)g(i,j)
\]  

The objective fitness function B \((0 \leq B \leq 1)\) is related to the trade-off between sensitivity (SV) and specificity (SP) needed on detection algorithms [Benson et al., 2000].

\[
B = 1 - \frac{\sqrt{(1-SP)^2 + (1-SV)^2}}{\sqrt{2}}; \text{where} SV = \frac{TP}{TP + FN} \text{ and } SP = \frac{TN}{FP + TN}
\]  

TP : number of true positives,  
TN : number of true negatives  
FP : number of false positives,  
FN : number of false negatives

For this experiment the goal is to convert the original image \((O)\) into the goal image\((G)\). If the pixel \(G(x_i, y_i) = 1\) then is

- a true positive if \(O(x_i, y_i) = 1\);
- a false positive if \(O(x_i, y_i) = 0\).
If the pixel $G(x,y) = 0$ then is

- a true negative if $O(x,y) = 0$;
- a false negative if $O(x,y) = 0$.

3.3.4. Terminal Set

We use a function set including two functions namely TRYEVAL1 and TRYEVAL2 of arity 1 and 2 respectively. The terminal set includes embedded strings

$$x(yz|w|)$$

where

- $x \in \{e, d\}$, represents the morphological operators (erosion and dilation);
- $y \in \{R, I\}$, represents the type of structuring element selected (regular and irregular);
- $z \in \{3, 5\}$, represents the size of structuring element and
- $w \in \{1...5\}$ represents the structuring element index

In fig. 3.6 is exemplified how the obtained parameters are transformed to linear representation. The obtained strings are straightforward used as MM algorithms (read left to right) to be applied over the training set during the evolution process. The GA features are those suggested for a simple GP in the literature [Koza, 1992]. Population of 50 parameters over 100 generations (to limit the computational effort) was used. A 0.9 crossover rate and 0.1 mutation rate using a half and half initialization method [Koza, 1992] was applied. It is important to remark that the aim of these experiments is to look for interesting patterns in the target image rather than performing a thorough comparison of performance.
3.4. Experimental Results of MM algorithm

We made a total of 100 GA runs combining 3 features to extract (heads, hooks and lines), 2 different numbers of training set elements (1 and 5), 2 different fitness functions, 2 different training set image sizes (16 x 16 and 32 x 32), 2 different combinations of structuring element types (R and I) and 3 different combinations of structuring elements sizes (3' and 5). Although the visual analysis made was not exhaustive (i.e. we focus on those MM algorithms with a good fitness value or with interesting string features) we made some interesting observations.

The fitness function B is more accurate than the fitness function A. The second has tendency to maximize either sensitivity or specificity. In our approach we avoid using human expertise according to the problem at hand (i.e. the three different...
MM feature extraction algorithms were obtained using GA in the same way. A possibility could be to use the fitness described in [Poli, 1996], but note that in such a case expert knowledge will be required for fitness function tuning.

GA registered different combinations of parameters to solve the heads extraction problem. The degree of accuracy is very difficult to evaluate visually and depending on the evaluator it does not always match the numerical fitness values obtained. However, we believe that many of the results obtained are as good as those that could be obtained by an expert writing the algorithm by hand. Fig.3.7 shows an image for visual inspection and shows the result of an MM algorithm generated by GA. The hooks extraction problem is fairly difficult task. That feature could be mismatched with lines, heads or other features present in a musical sheet. In spite of that, some GA generated algorithms present good approximations to the desired task.

Contrary to expectation, the lines extraction problem was the one presenting the most difficulties for the GA approach proposed. The results obtained were either tending to completely white images or to mismatch the lines with other features. GA accurately finds the lines, but also includes most of the hooks present in the test image.

The results were improved when combining structuring elements of different sizes and types. The irregular kernels are very useful as complement but are poor in performance when used by them. It should be noted that small structuring elements obtain better performance when used alone. It should be preferred to use a small number (i.e. the 1-5 training sets obtained better results than the 15-25). The size of the images which was restricted due to the heavy computation, used in the training
set did not affect the final results much, except perhaps 16 x 16 cases was slightly better.

Figure 3.7 Example of testing image (LHS) and a good visual result for heads on this test image (RHS)
3. 5. Semantic Image Segmentation

3. 5. 1. Image understanding

Context plays a significant role in image understanding. Algorithms already discussed above are constructed using the basic MM operators, i.e. erosion and dilation to find procedures to convert a binary image into another containing just a particular characteristic of interest. This section presents a higher level extension of region growing methods with regard to image segmentation. Generally image segmentation algorithms merge regions on the basis of general heuristics using local properties of regions referred to as syntactic information based methods. Conversely, semantic information representing higher level knowledge was first used by Feldman in 1974. Including more information, especially information about assumed region interpretation, can help the merging process during image segmentation. Context and criteria for global optimization of region interpretation consistency also play an important role. We assume that regions have been detected in an image that correspond to objects or other image entities, and let the objects and their inter-relationships be described by a region adjacency graph and/or a semantic net. Object properties are described by unary relations, and inter-relationships between objects are described by binary (or n-ary) relations. The goal of scene labeling is to assign a label (a meaning) to each image object to achieve an appropriate image interpretation. The resulting interpretation should correspond with available scene knowledge. The labeling should be consistent, and should favor more probable interpretations if there is more than one option. Consistency means that no two objects of the image appear in an illegal configuration - e.g. an object labeled house in the middle of an object labeled lake will be considered inconsistent in most scenes.
Conversely, an object labeled *house* surrounded by an object labeled *lawn* in the middle of a *lake* may be fully acceptable.

Two main approaches are chosen to achieve this goal.

- **Discrete** labeling allows only one label to be assigned to each object in the final labeling. Effort is directed to achieving a consistent labeling all over the image.

- **Probabilistic** labeling allows multiple labels to co-exist in objects. Labels are probabilistically weighted, with a label confidence being assigned to each object label.

The main difference is in interpretation robustness. Discrete labeling always finds either a consistent labeling or detects the impossibility of assigning consistent labels to the scene. Often, as a result of imperfect segmentation, discrete labeling fails to find a consistent interpretation even if only a small number of local inconsistencies is detected. Probabilistic labeling always gives an interpretation result together with a measure of confidence in the interpretation. Even if the result may be locally inconsistent, it often gives a better scene interpretation than a consistent and possibly very unlikely interpretation resulting from a discrete labeling. Note that discrete labeling may be considered a special case of probabilistic labeling with one label probability always being 1 and all the others being 0 for each object.

The scene labeling problem (Fig. 3.8) is specified by:

- A set of objects $R_i; i=1,\ldots,N$.
- A finite set of labels $\Theta_i$ for each object $R_i$.
- A finite set of relations between objects.

55
• The existence of a compatibility function (reflecting constraints) between interacting objects.

To solve the labeling problem considering direct interaction of all objects in an image is computationally very expensive and approaches to solving labeling problems are usually based on constraint propagation. This means that local constraints result in local consistencies (local optima), and by applying an iterative scheme the local consistencies adjust to global consistencies (global optima) in the whole image.

Figure 3.8 Local configuration of objects in an image - part of a region adjacency graph

3.5.2. Representation of Image Regions

• region adjacency graph
• dual graph

The region adjacency graph is one in which costs are associated with both nodes and arcs, implying that an update of these costs must be included in the given algorithm as node costs change due to the connecting two regions $R_i$ and $R_j$. 
Algorithm 3.2 for updating the region adjacency graph and its dual to merge two regions

1. Region Adjacency Graph
   a) Add all non-existing arcs connecting region \( R_i \) and all regions adjacent to \( R_j \)
   b) Remove the node \( R_j \) and all its arcs from the graph

2. Dual Graph
   a) Remove all arcs corresponding to the boundaries between regions \( R_i \) and \( R_j \), from the graph
   b) For each node associated with these arcs:
      - If the number of arcs associated with the node is equal to 2, remove this node and combine the arcs into a single one
      - If the number of arcs associated with the node is larger than 2, update the labels of arcs that correspond to parts of borders of region \( R_j \) to reflect the new region label \( R_i \)
3. 5. 3. Semantic Region Growing

Consider remotely sensed photographs, in which regions can be determined with interpretations such as field, road, forest, town, etc. Adjacent regions with the same interpretation can be merged. The problem is that the interpretation of regions is not known and the region description may give unreliable interpretations. Then, it is natural to incorporate context into the region merging using a priori knowledge about relations (unary, binary) among adjacent regions, and then to apply constraint propagation to achieve globally optimal segmentation and interpretation throughout the image. A region merging segmentation scheme is now considered in which semantic information is used in later steps, with the early steps being controlled by general heuristics. Only after the preliminary heuristics have terminated are semantic properties of existing regions evaluated and further region merging is either allowed or restricted.

Algorithm 3.3 for Semantic region growing

1. Initialize a segmentation with many small regions.

2. Merge all adjacent regions that have at least one weak edge on their common boundary.

3. For preset constants $c_1$ and $c_2$ and threshold $T_1$, merge neighbouring regions $R_i$ and $R_j$ if $S_y \leq T_1$, where

$$S_y = \frac{c_i + a_y}{c_2 + a_y},$$

$$a_y = \frac{(area_i)^2 + (area_j)^2}{perimeter_i \cdot perimeter_j} \quad (3.5.1)$$

4. For all adjacent regions $R_i$ and $R_j$, compute the conditional probability $P$ that their mutual border $B_{ij}$ separates them into two
regions of the same interpretation \((\theta_i = \theta_j)\), equation (3.5.4). Merge regions \(R_i\) and \(R_j\) if \(P\) is larger than a threshold \(T_1\). If no two regions can be so merged continue with step (5).

5. For each region \(R_i\), compute the initial conditional probabilities

\[
P(\theta_i = \omega_k \mid X_i), \quad k = 1, \ldots, R
\]

(3.5.2)

6. Repeat this step until all regions are labeled as final. Find a non-final region with the highest confidence \(C_i\) in its interpretation (equation 3.5.7); label the region with this interpretation and mark it as final. For each non-final region \(R_j\) and each of its possible interpretations \(\omega_k, k = 1, \ldots, R\) update the probabilities of its interpretations according to equation (3.5.8)

The final two steps, where semantic information has been incorporated, represent a variation of a serial relaxation algorithm combined with a depth-first interpretation tree search.

The goal is to maximize an objective function

\[
F = \prod_{i,j=1,..R} P(B_{ij} \text{ is between } \theta_i, \theta_j \mid X(B_{ij})) \prod_{i=1,..R} P(\theta_i \mid X_i) \prod_{j=1,..R} P(\theta_j \mid X_j)
\]

(3.5.3)

The probability that a border \(B_{ij}\) between two regions \(R_i\) and \(R_j\) is a false one must be found in step (4). This probability \(P\) can be found as a ratio of conditional probabilities;

let \(P_i\) denote the probability that the boundary should remain,

let \(P_f\) denote the probability that the boundary is false (i.e. should be removed and the regions should be merged)
let $X(B_y)$ denote properties of the boundary $B_y$

Then

$$P = \frac{P_f}{P_f + P_i}$$

(3.5.4)

Where

$$P_f = \sum_{k=1}^{R} P(\theta_i = \theta_j \mid X(B_y))P(\theta_i = \omega_k \mid X_i)P(\theta_j = \omega_k \mid X_j)$$

(3.5.5)

$$P_i = \sum_{k=1}^{R} \sum_{i=1}^{R} P(\theta_i = \omega_k \text{ and } \theta_j = \omega_i \mid X(B_y))P(\theta_i = \omega_k \mid X_i)P(\theta_j = \omega_i \mid X_j)$$

(3.5.6)

The confidence $C_i$ of interpretation of the region $R_i$ (step (6)):

Let $\theta_i^1, \theta_i^2$ represent the two most probable interpretations of region $R_i$.

Then

$$C_i = \frac{P(\theta_i^1 \mid X_i)}{P(\theta_i^2 \mid X_i)}$$

(3.5.7)

After assigning the final interpretation $\theta_f$ to a region $R_f$, interpretation probabilities of all its neighbors $R_j$ (with non-final labels) are updated to maximize the objective function

$$P_{\text{new}}(\theta_j) = P_{\text{old}}(\theta_j)P(B_f \text{ is between regions labelled } \theta_f, \theta_j \mid X(B_y))$$

(3.5.8)

The computation of these conditional probabilities is very expensive in terms of time and memory. It may be advantageous to precompute them beforehand and refer to table values during processing; this table must have been constructed with suitable sampling. Appropriate models of the inter-relationship between region interpretations, the collection of conditional probabilities, and methods of confidence evaluation must be specified to implement this approach.
3. 6. Genetic Image Interpretation

The previous section described the first historical semantic region growing method, which is still conceptually up to date. There is a fundamental problem in the region growing segmentation approach - the results are sensitive to the split/merge order. The conventional split-and-merge approach usually results in an under-segmented or an over-segmented image. It is practically impossible to stop the region growing process with a high confidence that there are neither too many nor too few regions in the image. Region growing can be designed so that it always results in an over-segmented image and post-processing steps can be used to remove false boundaries. False over-segmented regions can be found in watershed segmentation. Conventional region growing approaches are based on evaluation of homogeneity criteria and the goal is either to split a non-homogeneous region or to merge two regions, which may form a homogeneous region. Result is sensitive to the merging order - even if a merge results in a homogeneous region, it may not be optimal. There is no mechanism for seeking the optimal merges. The semantic region growing approach to segmentation and interpretation starts with an over-segmented image in which some merges were not best possible. The semantic process is then trying to locate the maximum of some objective function by grouping regions which may already be incorrect and is therefore trying to obtain an optimal image interpretation from partially processed data where some significant information has already been lost. Conventional semantic region growing merges regions in an interpretation level only and does not evaluate properties of newly merged regions. It also very often ends in a local optimum of region labeling; the global optimum is not found because
of the character of the optimization. This results in unreliability of image segmentation and interpretation of complex images.

The genetic image interpretation method solves these basic problems in the following manner:

- Both region merging and splitting is allowed; no merge or split is ever final, a better segmentation is looked for even if the current segmentation is already good.

- Semantics and higher level knowledge are incorporated into the main segmentation process, not applied as post-processing after the main segmentation steps are over.

- Semantics are included in an objective evaluation function (that is similar to conventional semantic-based segmentation).

- In contrast to conventional semantic region growing, any merged region is considered a contiguous region in the semantic objective function evaluation and all its properties are measured.

- The genetic image interpretation method does not look for local maxima; its search is likely to yield an image segmentation and interpretation specified by a (near) global maximum of an objective function

The genetic image interpretation method is based on a ‘hypothesize and verify’ principle. An objective function which evaluates the quality of a segmentation and interpretation is optimized by a genetic algorithm. The method is initialized with an over-segmented image called a primary segmentation, in which starting regions are called primary regions. Primary regions are repeatedly merged into current regions during the segmentation process. The genetic algorithm is
responsible for generating new populations of feasible image segmentation and interpretation hypotheses. Genetic algorithms test the whole population of segmentations, the better segmentations survive, and others die. If the objective function suggests that some merge of image regions was a good merge, it is allowed to survive into the next generation of image segmentation (the code string describing that particular segmentation survives), while bad region merges are removed (their description code strings die). The primary region adjacency graph is the adjacency graph describing the primary image segmentation. The specific region adjacency graph represents an image after the merging of all adjacent regions of the same interpretation into a single region (collapsing the primary region adjacency graph. The genetic algorithm requires any member of the processed population to be represented by a code string. Each feasible image segmentation defined by a generated code string (segmentation hypothesis) corresponds to a unique specific region adjacency graph. The specific region adjacency graphs serve as tools for evaluating objective segmentation functions. Design of a segmentation optimization function (the fitness function in genetic algorithms), is crucial for a successful image segmentation. The conventional approach evaluates image segmentation and interpretation confidences of all possible region interpretations. Based on the region interpretations and their confidences, the confidences of neighboring interpretations are updated, some being supported, and others becoming less probable. This conventional method can easily end at a consistent but sub-optimal image segmentation and interpretation. In the genetic approach, the algorithm is fully responsible for generating new and increasingly better hypotheses about image segmentation. Only these hypothetical segmentations are evaluated by the objective
function. Another significant difference is in the region property computation - as mentioned earlier, a region consisting of several primary regions is treated as a single region in the property computation process which gives a more appropriate region description.

Optimization criteria consist of three parts.

A confidence in the interpretation \( \theta_i \) of the region \( R_i \) according to the region properties \( X_i \),

\[
C(\theta_i | X_i) = P(\theta_i | X_i) \quad (3.5.9)
\]

A confidence in the interpretation \( \theta_i \) of a region \( R_i \) according to the interpretations \( \theta_j \) of its neighbors \( R_j \),

\[
C(\theta_i) = \frac{C(\theta_i | X_i) \sum_{j=1}^{N_A} (r(\theta_i, \theta_j) C(\theta_j | X_j))}{N_A} \quad (3.5.1)
\]

where \( r(\theta_i, \theta_j) \) represents the value of a compatibility function of two adjacent objects \( R_i \) and \( R_j \) with labels \( \theta_i \) and \( \theta_j \); \( N_A \) is the number of regions adjacent to the region \( R_i \).

An evaluation of interpretation confidences in the whole image

\[
C_{image} = \frac{\sum_{i=1}^{N_R} C(\theta_i)}{N_R} \quad (3.5.11)
\]

or

\[
C_{image}^f = \sum_{i=1}^{N_R} \left( \frac{C(\theta_i)}{N_R} \right)^2 \quad (3.5.12)
\]
where $N_R$ is the number of regions in the corresponding specific region adjacency graph.

The genetic algorithm attempts to optimize the objective function $C_{image}$, which represents the confidence in the current segmentation and interpretation hypothesis. As presented, the segmentation optimization function is based on both unary properties of hypothesized regions and on binary relations between these regions and their interpretations. The method is described by the following algorithm:

**Algorithm 3.4 for Genetic Image Segmentation and Interpretation**

1. Initialize the segmentation into primary regions, and define a correspondence between each region and the related position of its label in the code strings generated by the genetic algorithm.

2. Construct a primary region adjacency graph.

3. Pick the starting population of code strings at random. If a priori information is available that can help to define the starting population, use it.

4. Genetic Optimization: Collapse a region adjacency graph for each code string of the current population (Algorithm 3.2).

5. If the maximum of the optimization criterion does not increase significantly in several consecutive steps, go to step(7).

7. The code string with the maximum confidence (the best segmentation hypothesis) represents the final image segmentation and interpretation.

3.6.1. Case Study: Brain MR Image segmentation

The genetic algorithm has to work with larger string populations in practical applications which require a more complex priori knowledge, the primary image segmentation has more regions, and the optimum solution is not found in a few steps. Nevertheless the principles remain the same even when the method is applied to more complex problems. Interpretation of human magnetic resonance brain images is given here as such a complex example. The genetic image interpretation method was trained on two-dimensional MR images depicting anatomically-corresponding slices of the human brain. Knowledge about the unary properties of the specified neuro-anatomic structures and about the binary properties between the structure pairs was acquired from manually traced contours in a training set of brain images. The unary region confidences \( C(\theta_i \mid X_i) \) and the compatibility functions \( r(\theta_i, \theta_j) \) were calculated based on the brain anatomy and MR image acquisition parameters.
Unary Confidences:

- The unary confidence of a region was calculated by matching the region's shape and other characteristic properties with corresponding properties representing the hypothesized interpretation.

- Let the set of properties of region $R_i$ be
  \[ X_i = \{x_{i1}, x_{i2}, \ldots, x_{iN} \} \]

- Matching was done for each characteristic of the region $\{x_j\}$, and the unary confidence $C(\theta_i \mid X_i)$ was calculated as follows:
  \[ C(\theta_i \mid X_i) = P(x_{i1}) \cdot P(x_{i2}) \cdot \ldots \cdot P(x_{iN}) \] (3.6)

The feature confidences $P(x_{ik})$ were calculated by using the piecewise linear function as shown in Fig 3.10.

For example, let $x_{ik}$ be the area of region $R_i$ in the specific RAG and let $R_i$ be labeled $\theta_i$. According to a priori knowledge, assume that an object labeled $\theta_i$ has an area $y_{ik}$. Then

\[ \bar{x} = (\bar{x}_1, \bar{y}_1, \bar{x}_2, \bar{y}_2, \ldots, \bar{x}_N, \bar{y}_N) \]

The limit L depends on the strength of the a priori knowledge for each particular feature.

Binary confidences:

The value of the compatibility function $r(\theta_i, \theta_j)$ was assigned to be in the range $[0,1]$ depending on the strength of the a priori knowledge about the expected
configuration of regions $R_i$ and $R_j$. Low binary confidences serve to penalize infeasible configurations of pairs of regions. Similar to the calculation of the unary confidence, the compatibility function was calculated as a product of local binary relations. After the objective function $C_{image}$ was designed using a number of brain images from the training set, the genetic brain image interpretation method was applied to testing brain images.

![Feature confidence graph](image)

**Figure 3.10** Piecewise linear function for calculating unary confidences. L is a limit which depends on the a priori knowledge.

For illustration, the primary region adjacency graph typically consisted of approximately 400 regions; a population of 20 strings and a mutation rate $\mu = 1/\text{string\_length}$ was used during the genetic optimization. The method was applied to a testing set of MR brain images and offered good image interpretation performance.

### 3.6.2. Experimental Results

Conventional semantic region growing methods start with a non-semantic phase and use semantic post processing to assign labels to regions. Based on the
segmentation achieved in the region growing phases, the labeling process is trying to find a consistent set of interpretations for regions. The genetic image interpretation approach functions in a quite different way.

- Firstly, there are no separate phases - the semantics are incorporated into the segmentation/interpretation process.

- Secondly, segmentation hypotheses are generated first, and the optimization function is used only for evaluation of hypotheses.

- Thirdly, a genetic algorithm is responsible for generating segmentation hypotheses in an efficient way.

The method can be based on any properties of region description and on any relations between regions. The basic idea of generating segmentation hypotheses solves one of the problems of split-and-merge region growing - the sensitivity to the order of region growing. The only way to re-segment an image in a conventional region growing approach if the semantic post-processing does not provide a successful segmentation is to apply feedback control to change region growing parameters in a particular image part. There is no guarantee that a global segmentation optimum will be obtained even after several feedback re-segmentation steps. In the genetic image interpretation approach, no region merging is ever final. Natural and constant feedback is contained in the genetic interpretation method because it is a part of the general genetic algorithm - this gives a good chance that a (near) global optimum segmentation/interpretation will be found in a single processing stage.
Figure 3.11. Automated segmentation and interpretation of MR brain images. (a) Observer-defined borders of the neuro-anatomic structures correspond closely with (b) computer-defined borders.
We note that the image interpretation / understanding methods cannot and do not guarantee a correct segmentation - all the approaches try to achieve optimality according to the chosen optimization function. Therefore, a priori knowledge is essential to design a good optimization function. A priori knowledge is often included into the optimization function in the form of heuristics. In the GA method, it may affect the choice of the starting population of segmentation hypotheses that can affect computational efficiency.

Another important property of the GA method is the possibility of parallel implementation. This method is naturally parallel. Moreover, there is a straightforward generalization leading to a genetic image segmentation and interpretation in three dimensions. Considering a set of image planes forming a three-dimensional image (like MR or CT images), a primary segmentation can consist of regions in all image planes and can be represented by a 3D primary relational graph. The interesting possibility is to look for a global three-dimensional segmentation and interpretation optimum using 3D properties of generated 3D regions in a single complex processing stage. In such an application, the parallel implementation would be a necessity.

3. 7. Conclusion

This chapter so far explained our experiment of morphological binary image analysis based on Genetic Algorithm. We have applied this approach to the extraction of geometrical features in synthetic images as well as heads and hooks in musical score sheets obtaining promising results the details of which are discussed in chapter six.
Moreover another method of genetic image interpretation involving labeling of graphs and region growing methods have been explained and exploited using the MRI images of human brain. The genetic image interpretation method was trained on two-dimensional MR images depicting anatomically-corresponding slices of the human brain by deducing the unary properties and binary properties in a training set of brain images. This method has also proved to be successful provided the a-priori knowledge is stored in appropriate data structures.