Adaptive Image Segmentation Using a Genetic Algorithm

Bir Bhanu, *Senior Member, IEEE*, Sungkee Lee, *Member, IEEE*, and John Ming

Abstract—**Image segmentation is an old and difficult problem. One of the fundamental weaknesses of current computer vision systems to be used in practical applications is their inability to adapt the segmentation process as real-world changes occur in the image. We present the first closed loop image segmentation system which incorporates a genetic algorithm to adapt the segmentation process to changes in image characteristics caused by variable environmental conditions such as time of day, time of year, clouds, etc. The segmentation problem is formulated as an optimization problem and the genetic algorithm efficiently searches the hyperspace of segmentation parameter combinations to determine the parameter set which maximizes the segmentation quality criteria. The goals of our adaptive image segmentation system are to provide continuous adaptation to normal environmental variations, to exhibit learning capabilities, and to provide robust performance when interacting with a dynamic environment. We present experimental results which demonstrate learning and the ability to adapt the segmentation performance in outdoor color imagery.**

I. Introduction

IMAGE segmentation is typically the first, and most difficult, task (also known as an ill-defined problem) of any automated image understanding process. It refers to the grouping of parts of an image that have "similar" image characteristics. All subsequent interpretation tasks including object detection, feature extraction, object recognition, and classification rely heavily on the quality of the segmentation process. Despite the large number of segmentation techniques presently available [2], [10], [15], no general methods have been found that perform adequately across a diverse set of imagery. Only after numerous modifications to an algorithm's control parameter set can any current segmentation technique be used to process the wide diversity of images encountered in real world applications such as the operation of an autonomous robotic land vehicle or aircraft, automatic target recognizer, or a photointerpretation task.

When presented with an image from one of these application domains, selecting the appropriate set of algorithm parameters is the key to effectively segmenting the image [6]. The image segmentation problem can be characterized by several factors which make the parameter selection process very difficult. *First*, most of the powerful segmentation techniques available today contain numerous control parameters which must be adjusted to obtain optimal performance. As an example, the *Phoenix* segmentation algorithm [19], [26] used in our experiments contains 14 separate control parameters that directly affect the segmentation results. The size of the parameter search space in these systems can be prohibitively large, unless it is traversed in a highly efficient manner. *Second*, the parameters within most segmentation algorithms typically interact in a complex, non-linear fashion, which makes it difficult or impossible to model the parameters' behavior in an algorithmic or rule-based fashion. Thus, the multi-dimensional objective function defined using the various parameter combinations cannot generally be modeled in a mathematical way. *Third*, since variations between images cause changes in the segmentation results, the objective function that represents segmentation quality varies from image to image. The search technique used to optimize the objective function must be able to adapt to these variations between images. *Finally*, the definition of the objective function itself can be a subject of debate because there are no single, universally accepted measures of segmentation performance available with which to uniquely define the quality of the segmented image.

Hence, a need exists to apply an adaptive technique that can efficiently search the complex space of plausible parameter combinations and locate the values which yield optimal results. The approach should not be dependent on the particular application domain nor should it have to rely on detailed knowledge pertinent to the selected segmentation algorithm. Genetic algorithms (GAs), which are designed to efficiently locate an approximate global maximum in a search space, have the attributes described above and show great promise in solving the parameter selection problem encountered in the image segmentation task.

Fig. 1 illustrates the adaptive image segmentation task. Fig. 1(a) shows the original image that must be segmented by the system. Fig. 1(b) indicates the "ideal" segmentation of the image in which the wooden gates on either side of the road are properly segmented, along with the other main regions of the image (e.g., road, sky, fields, mountains, etc.). The genetic process converges on this ideal segmentation result through successive iterations. In Fig. 1(c), the image is grossly undersegmented. The next result, Fig. 1(d), has identified the main regions in the image, although the gate is still not evident. Fig. 1(e) oversegments the image by identifying too many irrelevant small regions. However, the wood gate regions have

Manuscript received August 27, 1993; revised July 15, 1994, and September 17, 1994. This work was supported in part by a Honeywell Initiative grant and ARPA/AFOSR grant F49620–93–1–0624. The content of the information does not necessarily reflect the position or the policy of the Government.

B. Bhanu is with the College of Engineering, University of California, Riverside, CA 92521.

S. Lee is with Kyungpook National University, Taegu, Korea 702–701.

J. Ming is with the AT&T Human Interface Technology Center, Atlanta, GA 30313.

IEEE Log Number 9414489.

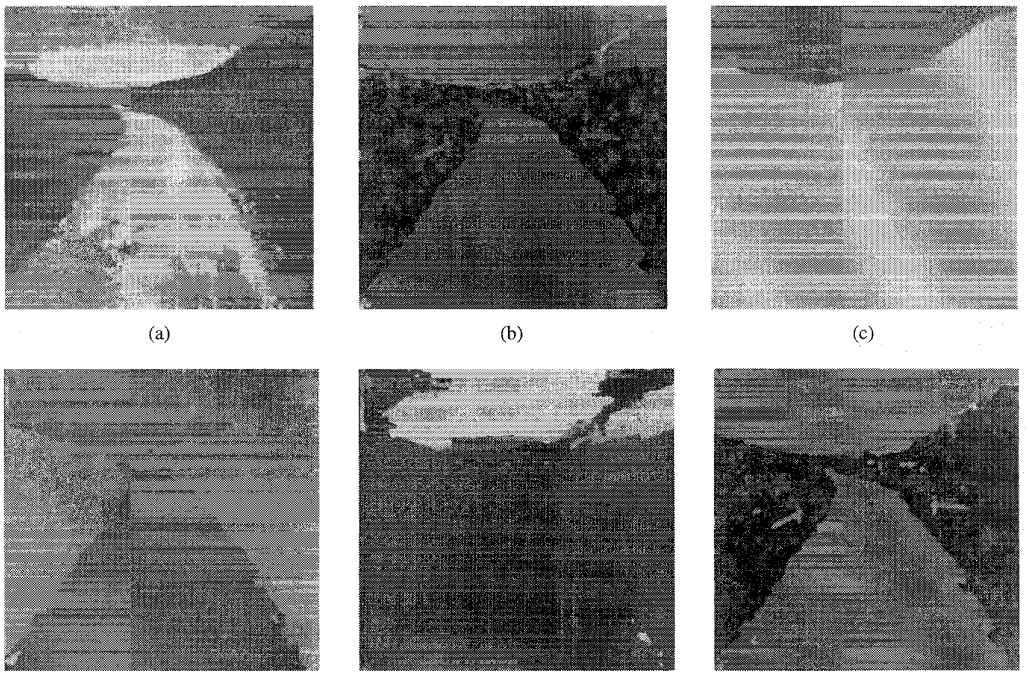


Fig. 1. Example of the adaptive image segmentation task. (a) Outdoor image, in which wooden gates adjacent to road must be segmented. (b) "Ideal" segmentation result obtained manually. (c) Initial, undersegmented result. (d) Second, more refined result which still lacks sufficient detail. (e) Third, oversegmented result. (f) Final result with gates properly extracted and other key image regions correctly segmented.

been obtained. The final result, shown in Fig. 1(f), eliminates most of the small regions while leaving the gate regions intact. This example illustrates the iterative, convergent nature of the genetic process towards ideal segmentation results.

The key elements of the adaptive image segmentation system described in this paper are:

- A closed-loop feedback control technique which provides an adaptive capability. The feedback loop consists of a genetic learning component, an image segmentation algorithm, and a segmented image evaluation component.
- A genetic learning system which optimizes segmentation performance on each individual image and accumulates segmentation experience over time to reduce the effort needed to optimize succeeding images.
- Image characteristics and external image variables are represented and manipulated using both numeric and symbolic forms within the genetic knowledge structure. Segmentation control parameters are represented and processed using a binary string notation.
- Image segmentation performance is evaluated using multiple measures of segmentation quality. These quality measures include *global* characteristics of the entire image as well as *local* features of individual object regions in the image. The global and local quality measures can be used separately or in combination.
- The adaptive segmentation system is very fundamental in nature and is not dependent on any specific segmentation algorithm or type of sensor data (visible, infrared, laser, etc.). The adaptive image segmentation system does not need to know the *inside details* of a segmentation algorithm except for the segmentation parameters and the range of values of these parameters so that they can be suitably represented in the genetic algorithms. The adaptive segmentation system adapts the segmentation parameters based on the quality of segmentation achieved using these parameter values. In this sense, the system is independent of the segmentation algorithm. There is nothing intrinsic to the system that is dependent on the type of images that need to be processed. As long as we can supply the quality measures for the segmented images, the adaptive segmentation system will optimize the quality measures. So, if there is an evaluation system for the segmented images, the adaptive system can be applied. However, a segmentation algorithm has a strong relationship with the type of images. In other words, it makes sense to apply a segmentation algorithm only to a particular class of images. For example, the *Phoenix* algorithm is useful for the segmentation of color images.

The focus of our work is not to develop yet another specialized segmentation algorithm that works only in a very limited domain on a few images, but is directed towards adapting the performance of a well known existing segmentation algorithm [19], [24], [26] across a wide variety of environmental conditions that cause changes in the image characteristics. While there are threshold selection techniques [22], [25], [28] that adapt to local image properties in an image for local image segmentation, these techniques do not adapt to changes in images caused by variations in the environmental conditions and do not accomplish any learning from experience to improve the performance of the system over

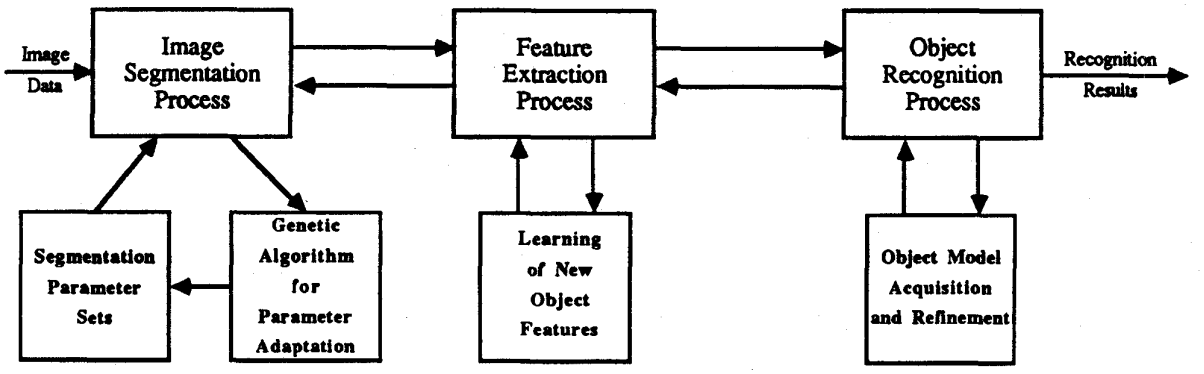


Fig. 2. Conceptual design of the multi-level computer vision process.

time. To date, no segmentation algorithm has been developed which can automatically generate an "ideal" segmentation result in one pass (or in an open loop manner) over a range of scenarios encountered in practical outdoor applications. Any technique, no matter how "sophisticated" it may be, will eventually yield poor performance if it cannot adapt to the variations in outdoor scenes. Therefore, in this paper we attempt to address this fundamental bottleneck in developing "useful" computer vision systems for practical scenarios by developing a closed-loop system that automatically adapts the segmentation algorithm's performance by changing its control parameters and will be valid across a wide diversity of image characteristics and application scenarios. It should be noted that the performance of the adaptive algorithm will be limited by the capabilities of the segmentation algorithm, but the results will be optimal for a given image based on our evaluation criteria.

Further, the adaptive image segmentation system presented in this work is designed to be a part of an overall approach to computer vision [3], [23] as shown in Fig. 2. The adaptive segmentation technique provides a segmented image that can be utilized at the intermediate level of the vision process to perform region labeling and feature extraction. Once this data is available, it is then passed to the object recognition stage where the objects of interest in the image are located and identified. By maximizing the segmentation quality at the lowest level of the vision process, we can increase the performance of the higher levels and improve the robustness of the overall vision process.

While it is true that the segmentation process shown in Fig. 2 (and described in this paper) is by itself a bottom-up segmentation process, some vision systems may require a top-down approach to image segmentation [21]. However, the objective in this work is to achieve the optimal segmentation of the image before the results are passed to higher-level processes. For example, if recognition or feature extraction results are unsatisfactory, a new segmentation algorithm can be used or the criteria for segmentation evaluation can be modified in this dynamic system. Although the segmentation and the interpretation processes are interlinked, in this paper we focus on improving the segmentation performance alone, without subjecting the adaptive segmentation process to the outcome of any higher-level interpretation process. Thus, this paper concentrates on the first part of Fig. 2, which is the adaptive image segmentation component.

The next section of this paper argues about the genetic algorithm (GA) as the appropriate search technique for the segmentation problem. Section III presents an overview of the GA-based approach, including previous applications in computer vision research. Following this review, Section IV describes the baseline adaptive image segmentation process that we have developed. We explain the choice of a particular segmentation algorithm as well as the manner in which segmentation quality is measured. Section V presents the experimental results on a sequence of outdoor images. It also describes different variations of the algorithm and discusses comparison results with other non-adaptive segmentation techniques. Finally, Section VI provides the conclusions of this paper.

II. Segmentation as an Optimization Problem

We previously highlighted some of the characteristics of the segmentation problem such as the size of the parameter search space, the complexity of the objective function, and variations in the objective function caused by changes in the imagery as well as the accepted definition of the function itself. Fig. 3(a) provides a generalized representation of an objective function that is typical for the image segmentation process. The figure depicts an application in which only two segmentation parameters are being varied, and the corresponding segmentation quality obtained for any pair of algorithm parameters. Because the algorithm parameters interact in complex ways, the objective function is multimodal and presents problems for many commonly used optimization techniques. Further, since the surface is derived from an analysis of real world imagery, it may be discontinuous, may contain significant amounts of noise, and cannot be described in closed form. Fig. 3(b) shows the actual segmentation quality surface derived for the outdoor image in Fig. 1(a). The derivation of this surface will be described in Section IV-D, where we discuss the segmentation evaluation process.

A. Selection of an Optimization Technique

The conclusion drawn from an analysis of the surfaces in Fig. 3 is that we must utilize a highly effective search strategy

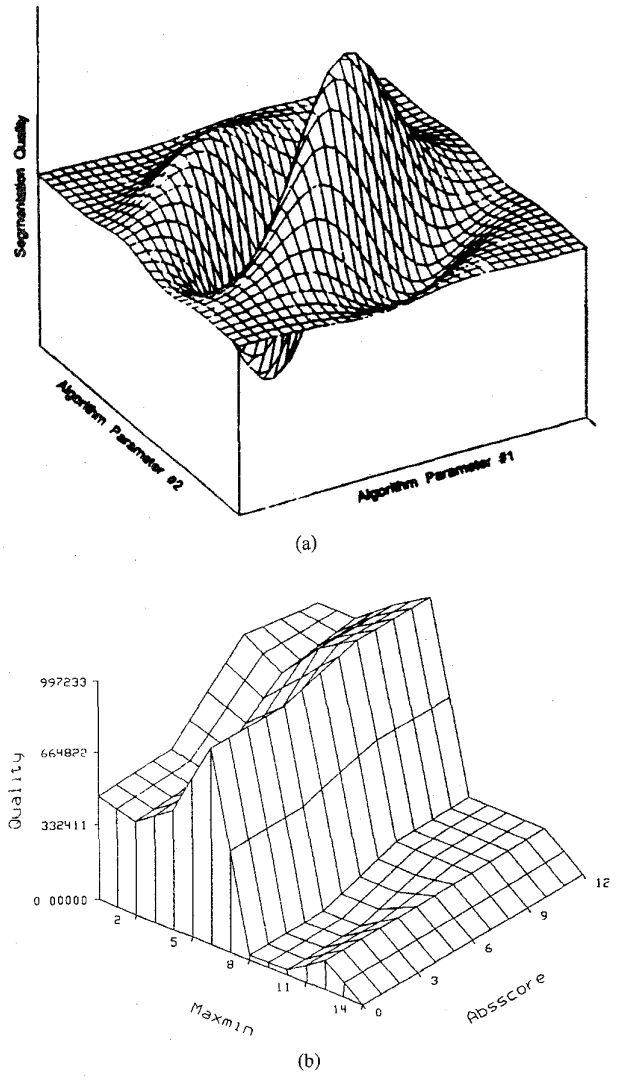


Fig. 3. (a) Representation of the objective function which must be optimized in the adaptive image segmentation problem. (b) Segmentation quality surface for the image shown in Fig. 1(a). The contours of the surface indicate the complex interactions between the control parameters (*Maxmin* and *Absscore* are *Phoenix* [19], [26] algorithm parameters).

which can withstand the breadth of performance requirements necessary for the image segmentation task. We have reviewed many of the techniques commonly used for function optimization to determine their usefulness for this particular task. In addition, we have also investigated other knowledge-based techniques which attempt to modify segmentation parameters using production rule systems. The drawbacks to each of these methodologies are as follows:

- *Exhaustive Techniques (Random walk, depth first, breadth first, enumerative)*—Able to locate global maximum but computationally prohibitive because of the size of the search space.
- *Calculus-Based Techniques (Gradient methods, solving systems of equations)*—No closed form mathematical representation of the objective function is available. Discontinuities and multimodal complexities are present in the objective function.
- *Partial Knowledge Techniques (Hill climbing, beam search, best first, branch and bound, dynamic programming, A^*)*—Hill climbing is plagued by the foothill, plateau, and ridge problems. Beam, best first, and A^* search techniques have no available measure of goal distance. Branch and bound requires too many search points while dynamic programming suffers from the *curse of dimensionality* and is expensive computationally.
- *Knowledge-Based Techniques (Production rule systems, heuristic methods)*—These systems have a limited domain of rule applicability, tend to be *brittle* [17], and are usually difficult to formulate. Further, the visual knowledge required by these systems may not be representable in knowledge-based formats.

Genetic algorithms are able to overcome many of the problems mentioned in the above optimization techniques. They search from a *population* of individuals (search points), which make them ideal candidates for parallel architecture implementation, and are far more efficient than exhaustive techniques. Since they use simple recombinations of existing high quality individuals and a method of measuring current performance, they do not require complex surface descriptions, domain specific knowledge, or measures of goal distance. Moreover, due to the generality of the genetic process, they are independent of the segmentation technique used, requiring only a measure of performance, which is referred to as segmentation quality, for any given parameter combination.

Genetic algorithms are also related to simulated annealing [7] where, although random processes are also applied, the search method should not be considered directionless. Both genetic algorithms and simulated annealing are modeled on processes found in nature (natural evolution and thermodynamics, respectively), and both techniques have recently attracted significant attention as suitable for optimization problems of very large scale. In the image processing domain, Geman and Geman [11], [18] have used simulated annealing to perform image restoration and Sontag and Sussmann [27] have performed image restoration and segmentation. Simulated annealing and other hybrid techniques [1] also have the potential for improved performance over earlier optimization techniques.

III. Overview of Genetic Algorithms

Genetic algorithms were pioneered at the University of Michigan by John Holland and his associates [8], [13], [16]. The term *genetic algorithm* is derived from the fact that its operations are loosely based on the mechanics of genetic adaptation in biological systems. Genetic algorithms can be briefly characterized by three main concepts: a Darwinian notion of fitness or strength which determines an individual's likelihood of affecting future generations through reproduction; a reproduction operation which produces new individuals by combining selected members of the existing population; and genetic operators which create new offspring based on the structure of their parents.

A genetic algorithm maintains a constant-sized *population* of candidate solutions, known as individuals. The initial seed population from which the genetic process begins can be

chosen randomly or on the basis of heuristics, if available for a given application. At each iteration, known as a *generation*, each individual is evaluated and recombined with others on the basis of its overall quality or *fitness*. The expected number of times an individual is selected for recombination is proportional to its fitness relative to the rest of the population. Intuitively, the high strength individuals selected for reproduction can be viewed as providers of "building blocks" from which new, higher strength offspring can be constructed. An abstract procedure of a simple genetic algorithm is given below, where $P(t)$ is a population of candidate solutions to a given problem at generation t.

```
t = 0;
initialize P(t);
evaluate P(t);
while not ⟨termination condition⟩
    begin
        t = t + 1;
        reproduce P(t) from P(t − 1);
        recombine P(t);
        evaluate P(t);
    end;
```

New individuals are created using two main genetic recombination operators known as *crossover* and *mutation*. Crossover operates by selecting a random location in the genetic string of the parents (crossover point) and concatenating the initial segment of one parent with the final segment of the second parent to create a new child. A second child is simultaneously generated using the remaining segments of the two parents. The string segments provided by each parent are the building blocks of the genetic algorithm. Mutation provides for occasional disturbances in the crossover operation by inverting one or more genetic elements during reproduction. This operation insures diversity in the genetic strings over long periods of time and prevents stagnation in the convergence of the optimization technique.

The individuals in the population are typically represented using a binary notation to promote efficiency and application independence of the genetic operations. Holland [16] provides evidence that a binary coding of the genetic information may be the optimal representation. Other characteristics of the genetic operators remain implementation dependent, such as whether both of the new structures obtained from crossover are retained, whether the parents themselves survive, and which other knowledge structures are replaced if the population size is to remain constant. In addition, issues such as the size of the population, crossover rate, mutation rate, generation gap, and selection strategy have been shown to affect the efficiency with which a genetic algorithm operates [14].

The inherent power of a genetic algorithm lies in its ability to exploit, in a highly efficient manner, information about a large number of individuals. By allocating more reproductive occurrences to above average individuals, the overall net affect is an upward shift in the population's average fitness. Since the overall average moves upward over time, the genetic algorithm is a "global force" which shifts attention to productive regions (groups of highly fit individuals) in the search space. However, since the population is distributed throughout the search space, genetic algorithms effectively minimize the problem of converging to local maxima.

Since genetic algorithm rely on the accumulation of evidence rather than on domain dependent knowledge, genetic algorithms are ideal for optimization in applications where domain theories or other applicable knowledge is difficult or impossible to formulate. However, there are certain drawbacks to genetic algorithms which make them inappropriate for certain applications. For example, genetic systems usually require the evaluation of a large number of candidate solutions. In application domains where the evaluation process is expensive, the computational effort to perform numerous evaluations may be prohibitive. However, research by Fitzpatrick and Grefenstette [9] has shown that a simple statistical approximation to a complex evaluation process can allow genetic systems to effectively adapt in these situations and converge to an approximate global maxima.

To date, genetic algorithms have been applied to a wide diversity of problems including combinatorial optimization, gas pipeline operations, and machine learning [13]. With regards to computer vision applications, Mandava *et. al.* [20] have used genetic algorithms in solving the vision problem of image registration. In this work, the genetic system was used to select a set of transformation parameters which correctly align a pair of images. Genetic algorithms have also been used in computer vision for generating image domain feature detectors by Gillies [12].

IV. Adaptive Image Segmentation System

Genetic algorithms can be used in three different ways to provide an adaptive behavior within a computer vision system. The simplest approach is to allow the genetic system to modify a set of control parameters that affect the output of an existing computer vision program. By monitoring the quality of the resulting program output, the genetic system can dynamically change the parameters to achieve the best performance. A second approach allows the genetic component to modify the complex data structures within an algorithm or production rule system for a computer vison application. By modifying the control mechanism or *agenda* in an algorithm or the organization of data frames in a rule-based system, the genetic algorithm can bring about changes in the system's behavior. Finally, the most complex implementation of an adaptive computer vision system allows the genetic algorithm to actually make changes in the executable code of a program. In most of these cases, the adaptation involves changing the condition/action statements of the rules in a production system. Since almost every image segmentation algorithm contains parameters that are used to control the segmentation results, we have adopted the first strategy listed above.

Adaptive image segmentation requires the ability to modify control parameters in order to respond to changes that occur in the image as a result of varying environmental conditions. The block diagram of our approach to adaptive image segmentation is shown in Fig. 4. After acquiring an input image, the system analyzes the image characteristics and passes this informa-

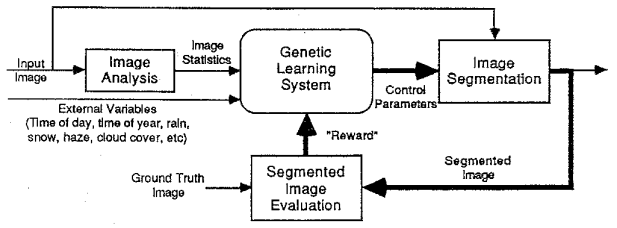


Fig. 4. Block diagram of the adaptive image segmentation process.

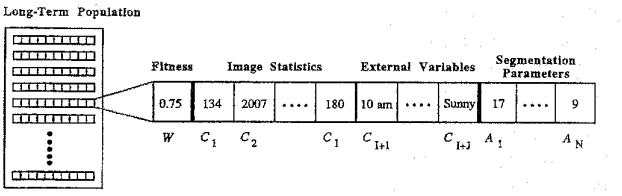


Fig. 5. Representation of a knowledge structure used by the genetic learning system. The image characteristics (image statistics and external variables), segmentation parameters, and the quality or *fitness* of the parameter set is stored in each knowledge structure.

tion, in conjunction with the observed external variables, to the genetic learning component. Using this data, the genetic learning system selects an appropriate parameter combination, which is passed to the image segmentation process. After the image has been segmented, the results are evaluated and an appropriate reward is generated and passed back to the genetic algorithm. This process continues until a segmentation result of acceptable quality is produced. The details of each component in this procedure will be described in the following subsections.

A. Image Characteristics

The input image must be analyzed so that a set of features can be extracted to aid in the parameter selection process performed by the genetic component. A set of characteristics of the image is obtained by computing specific properties of the digital image itself as well as by observing the environmental conditions in which the image was acquired. Each type of information encapsulates knowledge that can be used to determine a set of appropriate starting points for the parameter adaptation process.

Image analysis produces a set of image statistics that measure various properties of the digital image. There are a large number of plausible image statistics that can be used, including:

- **First Order Properties:** Measure the shape of the image histogram. Information includes mean, variance, skewness, kurtosis, energy, entropy, and x and y intensity centroids.
- **Second Order Properties:** Measure the histogram features based on joint probability distributions between pairs of pixels. Information includes autocorrelation, covariance, inertia, cooccurrence matrices, and other derived properties.
- **Histogram Peak/Valley Properties:** Measure the values of the peaks and valleys in the image histogram. Information includes maximum peak height divided by minimum valley height, total number of histogram peaks, maximum peak location, minimum valley location, distance between maximum peak and minimum valley, maximum peak-to-valley ratio, interval set score, and interval set size.

External variables can also be used to characterize an input image. These factors specify the conditions under which the image was acquired. They include information such as the time of day, time of year, cloud cover, temperature, humidity, and other environmental factors such as the presence of rain, snow, haze, fog, etc. These conditions all affect the quality of the image, which in turn necessitates changes in control parameters, and thus they provide useful information in representing the overall characteristics of the input image.

Fig. 5 illustrates the structure of the image statistics and external variables (C_i's) extracted from the image in Fig. 1(a). The image characteristics list used in our experiments is somewhat more complex than the one pictured in Fig. 5 since we are using color imagery. For the experiments described in Section V, we compute twelve first order properties for each color component (red, green, and blue) of the image. These features include mean, variance, skewness, kurtosis, energy, entropy, x intensity centroid, y intensity centroid, maximum peak height, maximum peak location, interval set score, and interval set size. The last two features measure histogram properties used directly by the *Phoenix* [19], [26] segmentation algorithm and provide useful image similarity information. Since we use a black/white version of the image to compute edge information and object contrast during the evaluation process, we also compute the twelve features for the Y (luminance component) image as well. Combining the image characteristic data from these four components yields a list of 48 elements. In addition, we utilize two external variables, time of day and weather conditions, in the outdoor experiments to characterize each image. The external variables are represented symbolically in the list structure (e.g., time = 9 am, 10 am, etc. and weather conditions = sunny, cloudy, hazy, etc.). The distances between these values are computed symbolically when measuring image similarity. The two external variables are added to the list to create an image characteristic list of 50 elements for the outdoor experiments.

B. Genetic Learning System

Once the image statistics and external variables have been obtained, the genetic learning component uses this information to select an initial set of segmentation algorithm parameters. A knowledge-based system is used to represent the image characteristics and the associated segmentation parameters. The knowledge structure (Fig. 5) stores the current fitness of the parameter settings, the image statistics and external variables of the image, and the segmentation parameter set used to process images with these characteristics. The image statistics and external variables form the condition portion of the knowledge structure, C_1 through C_{I+J}, while the segmentation parameters indicate the actions, A_1 through A_N, of the knowledge structure. The fitness, W, which ranges in value from 0.0–1.0, measures the quality of the segmentation

parameter set. Note that only the fitness value and the action portion of the knowledge structure are subject to genetic adaptation; the conditions remain fixed for the life of the knowledge structure.

When a new image is provided to the genetic learning system, the process begins by comparing the image characteristics of the new image with the knowledge structures in the global population (also called long-term population, Fig. 5). The global population represents the accumulated knowledge of the adaptive system obtained through previous segmentation experience. The algorithm computes a ranked list of individuals in the population that have characteristics similar to the new image. Ranking is based on the normalized Euclidean distance between the image characteristic values as well as the fitness of the knowledge structure. The normalized distance between images A and B is computed using

$$\text{dist}_{AB} = \sum_{i=1}^{I+J} W_i \left| \frac{C_{iA} - C_{i\text{MIN}}}{C_{i\text{MAX}} - C_{i\text{MIN}}} - \frac{C_{iB} - C_{i\text{MIN}}}{C_{i\text{MAX}} - C_{i\text{MIN}}} \right|.$$

where $C_{i\text{MIN}}$ is the minimum value of the ith numeric or symbolic feature in the global population, $C_{i\text{MAX}}$ is the maximum value of the ith feature in the global population, and W_i is the weight attached to the ith feature. In this work, the ranges are normalized and the W_i values have been set to 1 so that each feature contributes equally to the distance calculation.

When the distance between an image and several members of the global population are the same (e.g., if a previous image contributed multiple individuals to the global population), fitness values are used to select the best individuals from the population. Temporary copies of the highest ranked individuals are used to create the initial or seed population for the new image.

Once the initial or *seed* population is available, the genetic adaptation cycle begins. This cycle is shown in Fig. 6. (The seed population is the same as the initial population, when the genetic algorithm begins its search operation.) The segmentation parameter set in each member of the seed population is used to process the image. The quality of the segmented results for each parameter set is then evaluated. If the maximum segmentation quality for the current population is above a predefined threshold of acceptance or other stopping criteria are satisfied, the cycle terminates and the high quality members of the current image population are used to update the global population. Less fit members of the global population are discarded in favor of higher strength individuals obtained from processing the current image. In this manner, the system is able to extend the knowledge of the adaptive segmentation system by incorporating new experience into the knowledge database.

Alternatively, if after segmenting and evaluating the performance of the current or local (also called short-term) population, the system has not achieved acceptable segmentation quality and any other termination criteria are not satisfied, the genetic recombination operators are applied to the members of the current population. The crossover and mutation operators are applied to the high strength individuals in the population,

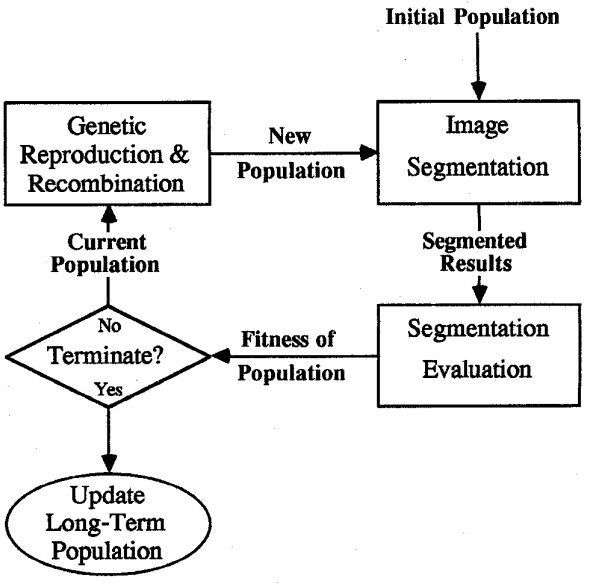


Fig. 6. Flow chart of genetic adaptation cycle. The cycle (segmentation-evaluation-reproduction-recombination) continues until termination criteria are satisfied. The long-term population is then modified in order to retain the information "learned" during the genetic adaptation process.

creating a new set of offspring which will theoretically yield better performance [16]. The new population is supplied back to the image segmentation process, where the cycle begins again. Each pass through the loop (segmentation-evaluation-recombination) is known as a generation. The cycle shown in Fig. 6 continues until the maximum fitness achieved at the end of a generation exceeds some threshold or other termination criteria are satisfied, as described earlier. The global population is updated and the system is then ready to process a new image.

C. Segmentation Algorithm

Since we are working with color imagery in our experiments, we have selected the *Phoenix* segmentation algorithm developed at Carnegie-Mellon University [19], [24], [26]. *Phoenix*, which was the subject of several Ph.D. dissertations, has been widely used, refined, and documented. The algorithm has been extensively tested on color imagery and has been assimilated into the DARPA/SRI Image Understanding Testbed [19].

The *Phoenix* algorithm is a recursive region splitting technique. An input image typically has red, green, and blue image planes, although monochrome images, texture planes, and other pixel-oriented data may also be used. Each of the data planes is called a feature or feature plane. The algorithm recursively splits nonuniform regions in the image into smaller subregions on the basis of a peak/valley analysis of the histograms of the red, green, and blue image components simultaneously. Fig. 7 shows a high-level description of the *Phoenix* segmentation process [19]. Segmentation begins with the entire image, considered to be a single region, based on histogram and spatial analyses. If the initial segmentation fails, the program terminates; otherwise, the program fetches each

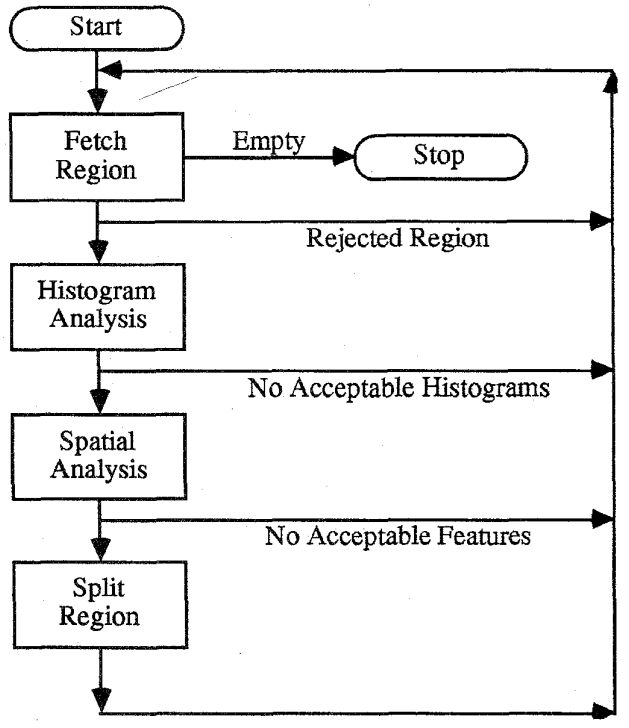


Fig. 7. Block diagram of the *Phoenix* [19], [26] algorithm.

of the new regions in turn and attempts to segment them. This process terminates when the recursive segmentation reaches a predefined *depth*, or when all the regions have been segmented as finely as various user-specified parameters permit.

Phoenix contains seventeen different control parameters, [19] fourteen of which are used to control the thresholds and termination conditions of the algorithm. There are about 10^{40} conceivable parameter combinations using these fourteen values. For the outdoor image sequence that we have used, these parameters can be divided into three groups according to their effect on segmentation results. Based on our experimentation, we find that of the fourteen values the two most critical parameters that affect the overall results of the segmentation process are, *maxmin* and *hsmooth*. From an analysis of the *Phoenix* algorithm, we find that incorrect values of these two parameters lead to results in which, at one extreme, the desired object is not extracted from the background, and at the other extreme, the object is broken up into many small regions that have little significance for higher-level processes. The default values for these parameters are 160 and 9, respectively. *Maxmin* specifies the lowest acceptable peak-to-valley-height ratio (of a histogram) used when deciding whether or not to split a large region into two or more smaller parts. *Hsmooth* controls the width of the window used to smooth the histogram of each image region during segmentation. Smoothing helps to remove small histogram peaks corresponding to small irrelevant regions or noise in the image. The use of only two parameters for the initial tests aids in the visualization of the optimization process since we can easily plot the associated segmentation quality corresponding to each parameter combination.

D. Segmentation Evaluation

After the image segmentation process has been completed by the *Phoenix* algorithm, we must measure the overall quality of the segmented image. There are a large number of segmentation quality measures that have been suggested in the literature [2], although none has achieved widespread acceptance as a universal measure of segmentation quality. In order to overcome the drawbacks of using only a single quality measure, we have incorporated an evaluation technique that uses five different quality measures to determine the overall fitness for a particular parameter set. Most of the individual measures of segmentation performance that we have selected for this work have been proposed in the computer vision literature and similar measures have been recommended by DARPA's Automatic Target Recognition Working Group (ATRWG) as good indicators of segmentation quality. In the following, boundary pixels refer to the pixels along the borders of the segmented regions, while the edges obtained after applying an edge operator are called edge pixels. The five segmentation quality measures that we have selected are,

1) *Edge-Border Coincidence*—Measures the overlap of the region borders in the image acquired from the segmentation algorithm relative to the edges found using an edge operator. In this quality measure, we use the Sobel operator to compute the necessary edge information. The original, unthinned Sobel edge image is used to maximize overlap between the segmented image and the edge image. Edge-border coincidence is defined as follows (refer to Fig. 8(a)). Let E be the set of pixels extracted by the edge operator after thresholding and S be the set of pixels found on the region boundaries obtained from the segmentation algorithm:

$$\begin{aligned} E &= \{p_1, p_2, \cdots, p_E\} \\ &= \{(x_{p1}, y_{p1}), (x_{p2}, y_{p2}), \cdots, (x_{pE}, y_{pE})\} \quad \text{and} \\ S &= \{q_1, q_2, \cdots, q_S\} \\ &= \{(x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \cdots, (x_{qS}, y_{qS})\}, \quad \text{then} \end{aligned}$$

$$\text{Edge-border Coincidence} = \frac{n(E \cap S)}{n(E)},$$

$$E \cap S = \{(x_k, y_k), k = 1, \cdots, m \quad \text{where} \quad (x_k, y_k \in E \text{ and } S\}, \quad \text{and}$$

$$n(A) = \text{the number of elements in set } A.$$

2) *Boundary Consistency*: Similar to edge-border coincidence, except that region borders which do not exactly overlap edges can be matched with each other. In addition, region borders which do not match with any edges are used to penalize the segmentation quality. The Roberts edge operator is used to obtain the required edge information. As with the edge-border coincidence measure, the Roberts edge image is not thinned to maximize the overlap between images. Boundary consistency is computed in the following manner (see Fig. 8(b)).

The first step is to find neighboring pixel pairs in the region boundary and edge results. For each pixel in the segmented image region boundary results, S, a neighboring pixel in the

edge image, E, that is within a distance of $d_{\max}$ is sought. A reward for locating a neighbor of the ith boundary pixel is computed using

$$R_i = \frac{d_{\max} - d_i}{d_{\max}},$$

where

$$d_{\max} = 10, \quad \text{and}$$
$$d_i = \text{the distance to the nearest edge pixel.}$$

Thus, if the pixels had overlapped, $R_i = (10 - 0)/10 = 1$. Pixels that do not directly overlap contribute a reward value that is inversely related to their distance from each other. As matching pairs of pixels are identified, they are removed from the region boundary and edge images (S and E). The total reward for all matching pixel pairs is obtained using

$$R_{\text{TOTAL}} = \sum_i R_i.$$

Once all neighboring pixel pairs have been removed from E and S, the remaining (i.e., non-overlapping and non-neighboring) pixels correspond to the difference between the two images. The average number of these pixels is used to compute a penalty

$$P = \frac{n(\text{all remaining pixels in } E \text{ and } S)}{2}.$$

Finally, since the value of boundary discrepancy must be positive, we define an intermediate value, M, as $M = (R_{\text{TOTAL}} - P)/n(E)$.
Then, Boundary Consistency $= M$, if $M \geq 0$, and zero otherwise.

3) *Pixel Classification*: This measure is based on the number of object pixels classified as background pixels and the number of background pixels classified as object pixels. Let G be the set of object pixels in the groundtruth image and R be the set of object pixels in the segmented image (see Fig. 8(c)). Formally, we have

$$G = \{p_1, p_2, \cdots, p_A\}$$
$$= \{(x_{p1}, y_{p1}), (x_{p2}, y_{p2}), \cdots, (x_{pA}, y_{pA})\} \quad \text{and}$$
$$R = \{q_1, q_2, \cdots, q_B\}$$
$$= \{(x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \cdots, (x_{qB}, y_{qB})\}.$$

Since pixel classification must be positive, we define the intermediate value N as follows

$$N = 1 - \left[\frac{(n(G) - n(G \cap R)) + (n(R) - n(G \cap R))}{n(G)}\right].$$

where

$$G \cap R = \{(x_k, y_k), k = 1, \cdots, m \text{ where } (x_k, y_k) \in G \text{ and } R\}.$$

Using the value of N, pixel classification can then be computed as

$$\text{Pixel Classification} = N, \text{ if } N \geq 0, \text{ and zero otherwise.}$$

4) *Object Overlap*: Measures the area of intersection between the object region in the groundtruth image and the segmented image, divided by the object region. As defined in the pixel classification quality measure, let G be the set of object pixels in the groundtruth image and R be the set of object pixels in the segmented image (Fig. 8(d)). Object overlap can be computed as

$$\text{Object Overlap} = \frac{n(G \cap R)}{n(G)},$$

where

$$G \cap R = \{(x_k, y_k), k = 1, \cdots, m \text{ where } (x_k, y_k) \in G \text{ and } R\}.$$

5) *Object Contrast*: Measures the contrast between the object and the background in the segmented image, relative to the object contrast in the groundtruth image. Let G be the set of object pixels in the groundtruth image and R be the set of object pixels in the segmented image, as shown in Fig. 8(a). In addition, we define a bounding box (X and Y) for each object region in these images. These boxes are obtained by enlarging the size of the minimum bounding rectangle for each object (G and R) by 5 pixels on each side. The pixels in regions X and Y include all pixels inside these enlarged boxes with the exception of the pixels inside the G and R object regions.

We compute the average intensity for each of the four regions (G, R, X, and Y) using the equation $I_L = \Sigma_{j=1}^{L_{\max}} I(j)/L_{\max}$, where $I(j)$ is the intensity of the jth pixel in some region L and $L_{\max}$ is the total number of pixels in region L. The contrast of the object in the groundtruth image, C_{GT}, and the contrast of the object in the segmented image, C_{SI}, can be computed using

$$C_{GT} = \left|\frac{I_G - I_X}{I_G}\right|, \quad C_{SI} = \left|\frac{I_R - I_Y}{I_R}\right|.$$

The object contrast quality measure is then computed as

$$\text{Object Contrast} = \frac{C_{SI}}{C_{GT}}, \text{ if } C_{GT} \geq C_{SI} \text{ or } \frac{C_{GT}}{C_{SI}}, \text{ if } C_{GT} < C_{SI}.$$

The maximum and minimum values for each of the five segmentation quality measures are 1.0 and 0.0, respectively. The first two quality measures are *global* measures since they evaluate the segmentation quality of the whole image with respect to edge information. Conversely, the last three quality measures are *local* measures since they only evaluate the segmentation quality for the object regions of interest in the image. When an object is broken up into smaller parts during the segmentation process, only the largest region which overlaps the actual object in the image is used in computing the local quality measures.

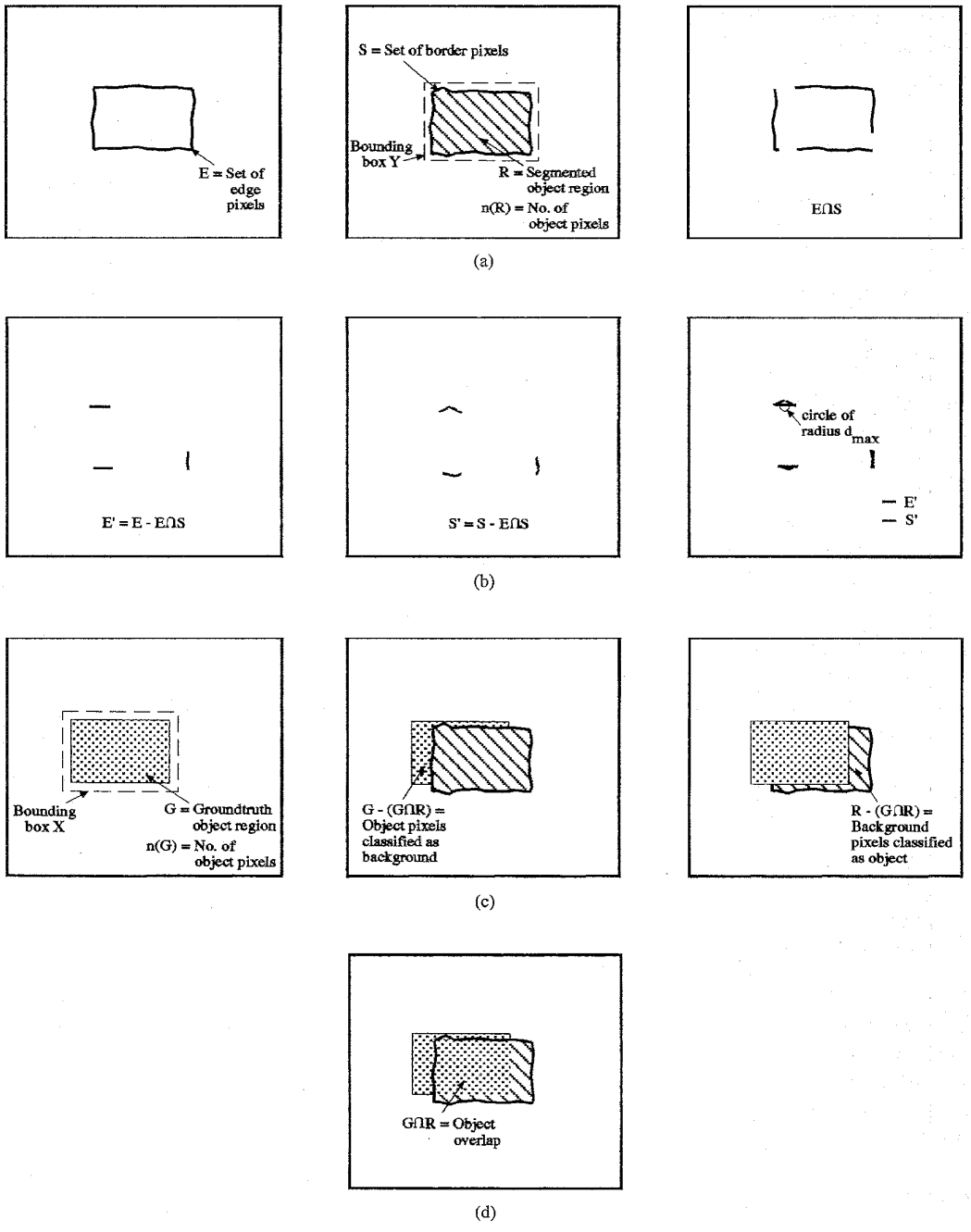


Fig. 8. Illustrations for the quality measures used in the adaptive segmentation system. (a) Edge-border coincidence diagram. (b) Boundary consistency diagram. (c) Pixel classification diagram. (d) Object overlap diagram.

The three local measures require the availability of object groundtruth information in order to correctly evaluate segmentation quality. Since object groundtruth data may not always be available, we have designed the adaptive segmentation system to use three separate methods of evaluating segmentation quality. First, we can measure quality using global evaluation methods alone. Second, if groundtruth data is available and we are only interested in correctly segmenting the object regions in the image, we can use local evaluation methods alone. Finally, if we desire good object regions as well as high quality overall segmentation results, we can combine global and local quality measures to obtain a *combined* segmentation quality measure that maximizes overall performance of the system.

In the experiments described in the next section, we combine the five quality measures into a single, scalar measure of segmentation quality using a weighted sum approach. Each of the five measures is given equal weighting in the weighted sum. In addition to the weighted sum technique currently

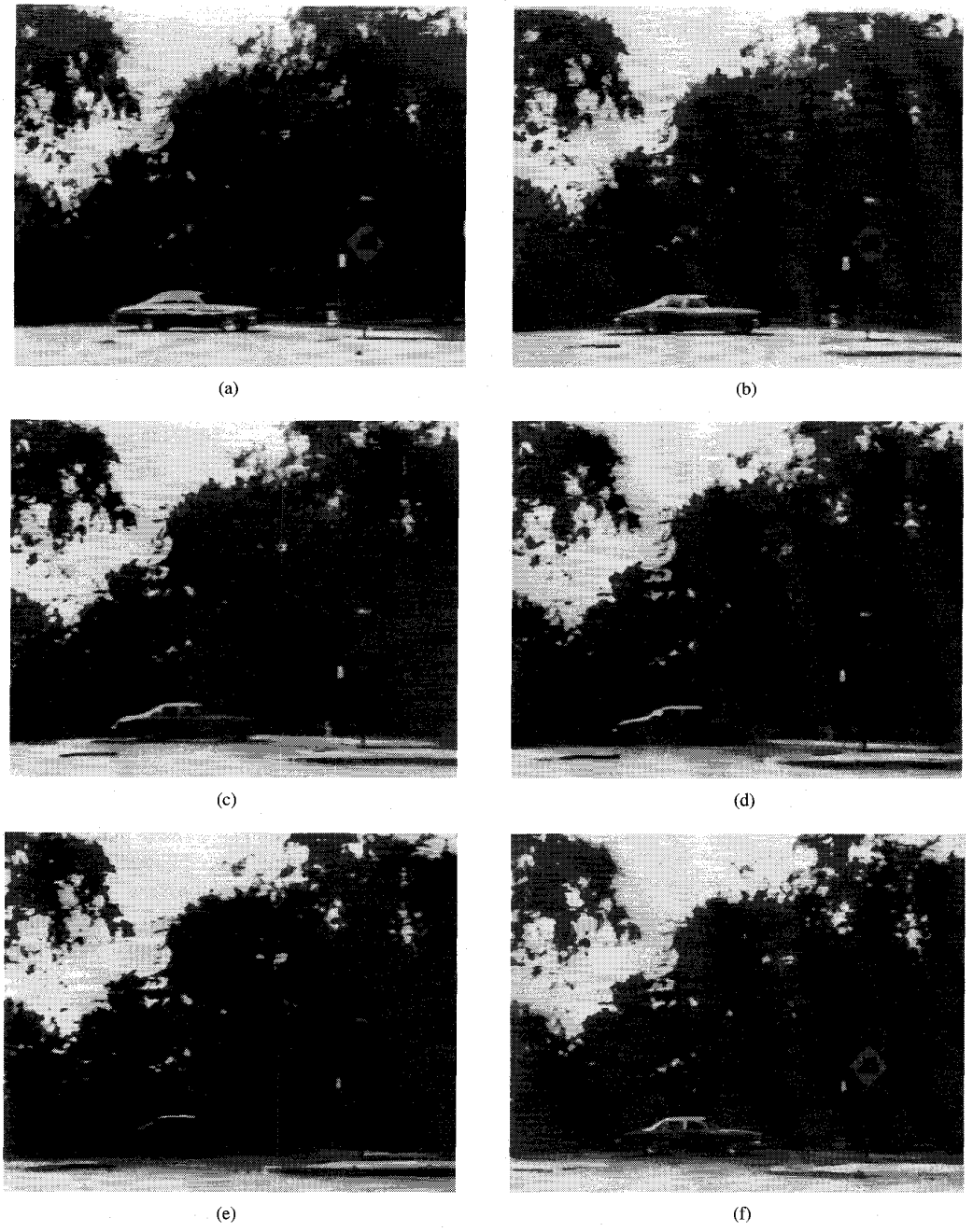


Fig. 9. Color images for the outdoor experiments. (a) Frame 1. (b) Frame 6. (c) Frame 10. (d) Frame 11. (e) Frame 13. (f) Frame 20.

in use, we have investigated a more complex vector evaluation approach that provides multidimensional feedback on segmentation quality [5].

V. Experimental Results

The adaptive image segmentation consists of the following steps:

1. Compute the image statistics.
2. Generate an initial population.
3. Segment the image using initial parameters.
4. Compute the segmentation quality measures.
5. WHILE not (stopping conditions) DO
   - 5a. select individuals using the reproduction operator
   - 5b. generate new population using the crossover and mutation operators
   - 5c. segment the image using new parameters
   - 5d. compute the segmentation quality measuers

   END
6. Update the knowledge base using the new knowledge structures.

We have tested the performance of the adaptive image segmentation system on a time sequence of outdoor images that contains variations in the position of the light source (sun) and the amount of light as well as changing environmental conditions. Imagery of this type allows us to monitor the system's ability to compensate for real world conditions. As the results will demonstrate, the adaptive segmentation technique is very effective in compensating for the changes observed in these images.

The outdoor image database consists of twenty frames captured using a JVC GXF700U color video camera. The images were collected approximately every 15 minutes over a 4 hour period. A representative subset of these images is shown in Fig. 9. The original images were digitized to be 480×480 pixels in size but were subsequently subsampled (average of 4×4 pixel neighborhood) to produce 120×120 pixel images for the segmentation experiments. Fig. 10 shows the time and associated weather conditions for each frame in the database. Frames 14, 17–19 were not collected at the 15-minute interval marks to ensure diversity in the environmental conditions within the outdoor database.

This type of image data simulates a photointerpretation scenario in which the camera position is fixed and the image undergoes significant change over time. Weather conditions in our image database varied from bright sun to overcast skies. Varying light level is the most prominent change throughout the image sequence. The changing environmental conditions caused by movement of the sun also created varying object highlights, moving shadows, and many subtle contrast changes between the objects in the image. Also, the colors of most objects in the image are subdued. The car in the image is the object of interest. The auto-iris mechanism in the camera was functioning, which causes a similar appearance in the background foliage throughout the image sequence. Notice that even with the auto-iris capability built into the camera, there is still a wide variation in image characteristics across the image sequence. This variation requires the use of an adaptive segmentation approach to compensate for these changes.

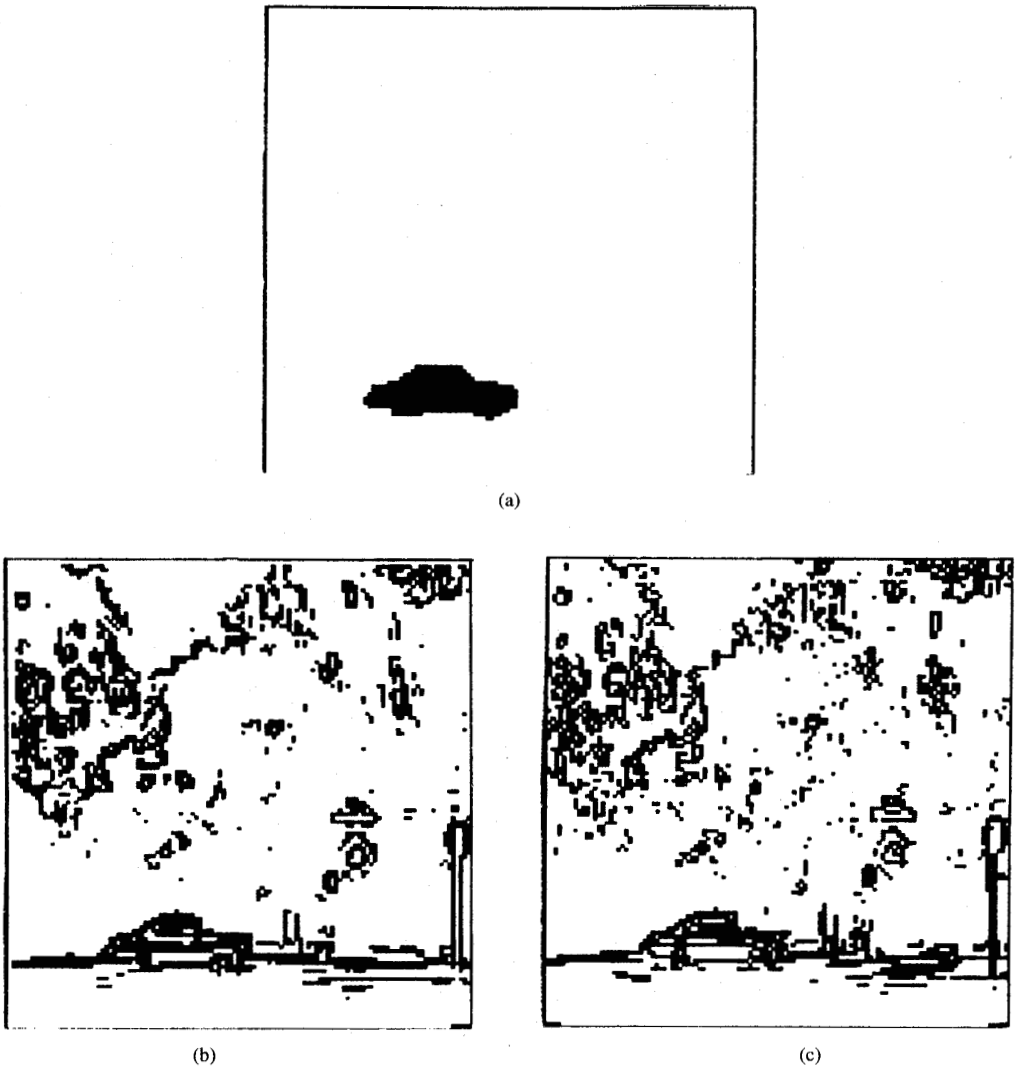
To precisely evaluate the effectiveness of the adaptive image segmentation system, we exhaustively defined the segmentation quality surfaces for each frame in the database. The car in the image is the object of interest for the pixel classification, object overlap, and object contrast segmentation quality measures. The groundtruth image for the car was obtained by manual segmentation of Frame 1 only for the image sequence and is shown in Fig. 11(a). The Sobel and Roberts edge operator results, which are used in the computation of the edge-border coincidence and boundary consistency measures respectively, are obtained from the black/white image (Y component of the YIQ image set) for each frame. The Sobel threshold value was 160 and the Roberts threshold value was 40 for the images. After applying edge operators to the first frame of the outdoor images, we took (manually) top 15% of pixels as edge pixels. Thus, we end up with a Sobel threshold value of 160 and Roberts threshold value of 40 for the outdoor images. These thresholds remain fixed for the remaining frames. So, in the remaining frames, edge pixels were not exactly in the top 15%. They were in the range of top 12–17%. If the thresholds are high, we will have less number of edge pixels and the segmentation results will be undersegmented, i.e., we will not have all the interesting regions. If the thresholds are too low, we will have more edge pixels and the segmentation results will be oversegmented, i.e., we will have too many small irrelevant regions. We have chosen thresholds so that segmentation results are neither undersegmented nor oversegmented. The Sobel and Roberts edge images for Frame 1 in Fig. 9 are shown in Fig. 11(b) and 11(c). For the determination of object contrast, we used 5 pixels beyond the Minimum Bounding Rectangle (MBR) for each object region. The *maxmin* and *hsmooth* parameters of the *Phoenix* algorithm were used to control the segmentation quality and the segmentation quality surfaces were defined for preselected ranges of these two parameters. All the parameters that were not optimized were set at the default *Phoenix* parameter values. These parameters remain fixed throughout all the experiments. *Maxmin* values, which affect segmentation performance in a non-linear fashion, were sampled exponentially over a range of values from 100 to 471. Values near 100 were spaced closer together than values at the upper end of the range. *Hsmooth* values were sampled linearly using numbers between 1 and 63. By selecting 32 discrete values (5 bits of resolution) for each of these parameter ranges, the search space contained 1024 different parameter combinations.

Frame #	Time	Weather
1	1:20 pm	Sunny
2	1:30 pm	Sunny
3	1:45 pm	Sunny
4	2:00 pm	Sunny
5	2:15 pm	Sunny
6	2:30 pm	Sunny
7	2:45 pm	Sunny
8	3:00 pm	Sunny
9	3:15 pm	Sunny
10	3:30 pm	Sunny
11	4:00 pm	Sunny
12	4:30 pm	Sunny
13	4:45 pm	Sunny
14	4:47 pm	Cloudy
15	5:00 pm	Sunny
16	5:15 pm	Sunny
17	5:20 pm	Cloudy
18	5:22 pm	Cloudy
19	5:25 pm	Sunny
20	5:30 pm	Cloudy

Fig. 10. Time of day and weather conditions for the outdoor images.

A. Basic Experiments

The first set of experiments with the adaptive segmentation system was divided into two separate phases: 1) a training



(a) (b) (c)

Fig. 11. Groundtruth data for the outdoor images. (a) Car region extracted from Frame 1 in Fig. 9. (b) Sobel edge results for Frame 1. (c) Roberts edge results for Frame 1.

phase where the optimization capabilities of the genetic algorithm were measured; and 2) a testing phase where we evaluated the reduction in effort achieved by utilizing previous segmentation experience. The outdoor image sequence was separated into two halves, 10 images for training and 10 images for testing. To insure that the training imagery was representative of the entire outdoor image sequence, the odd numbered images $(1, 3, \cdots, 19)$ were chosen as the training data and the even numbered images $(2, 4, \cdot, 20)$ were saved for testing purposes. During the training phase, seed populations were selected using random locations on the combined segmentation quality surface for each image. The genetic system was then invoked using the seed population for each image and the convergence rate of the process was measured. Each training image was processed 100 times, each with a different collection of random starting points. These results were combined to compute the average number of generations needed to optimize each surface. The genetic component used a local or seed population size of 10, a crossover rate of 0.8, and mutation rate of 0.01. A crossover rate of 0.8 indicates that, on average, 8 out of 10 members of the population will be selected for recombination during each generation. The mutation rate of 0.01 implies that on average, 1 out of 100 bits is mutated during the crossover operation to insure diversity in the local population.

The stopping criteria for the genetic process contains three tests. First, since the global maximum for each segmentation quality surface was known *a priori* (recall that the entire surface was precomputed), the first stopping criteria was the location of a parameter combination with 95% segmentation quality or higher. In experiments where the entire surface is not precomputed, this stopping criteria would be discarded. Second, the process terminates if three consecutive generations produce a decrease in the average population fitness for the local population. Third, if five consecutive generations fail to produce a new maximum value for the average population

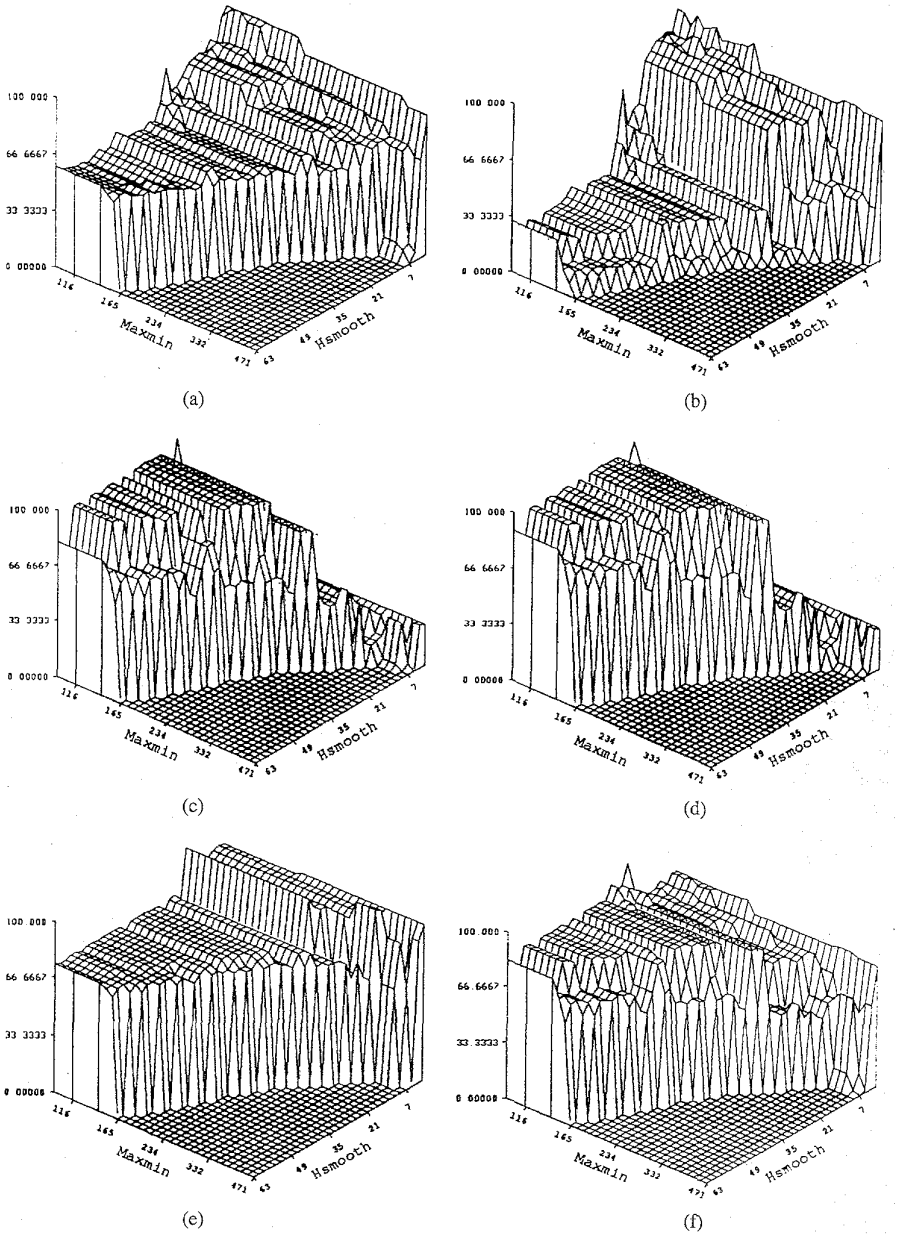


Fig. 12. Individual quality surfaces for Frame 1 in Fig. 9. (a) Edge-Border coincidence. (b) Boundary consistency. (c) Pixel clarification. (d) Object overlap. (e) Object contrast. (f) Combined segmentation quality.

fitness, the genetic process terminates. If any one of these three conditions is met, the processing of the current image is stopped and the maximum segmentation quality currently in the local population is reported.

Fig. 12 presents the five individual segmentation quality surfaces and the combined surface for Frame 1 of the database. The combined quality surfaces for the images of Fig. 9 are shown in Fig. 13. Notice that the individual surfaces in Fig. 12 as well as the combined surfaces in Fig. 13 are complex and hence, would pose significant problems to traditional optimization techniques. The maximum quality measure is totally dependent on the Phoenix system. The characteristic of the Phoenix segmentation algorithm is that when an image has low average brightness, Phoenix gives good segmentation results with low *maxmin* and *hsmooth* parameter values. Since the outdoor images have low average brightness in general, the maximum quality values are at the back corner of the surfaces.

Following the algorithm shown in Fig. 6, the average number of generations for each of the outdoor training images, as well as the average number of generations for all training images, is obtained. The maximum number of generations required by the genetic process was 13, the minimum number was 5, and the average number of generations was 9. The progression of the genetic search process for Frames 1, 11, and 13 from the starting parameter location to final location is shown in Fig. 14. By monitoring the intermediate generations

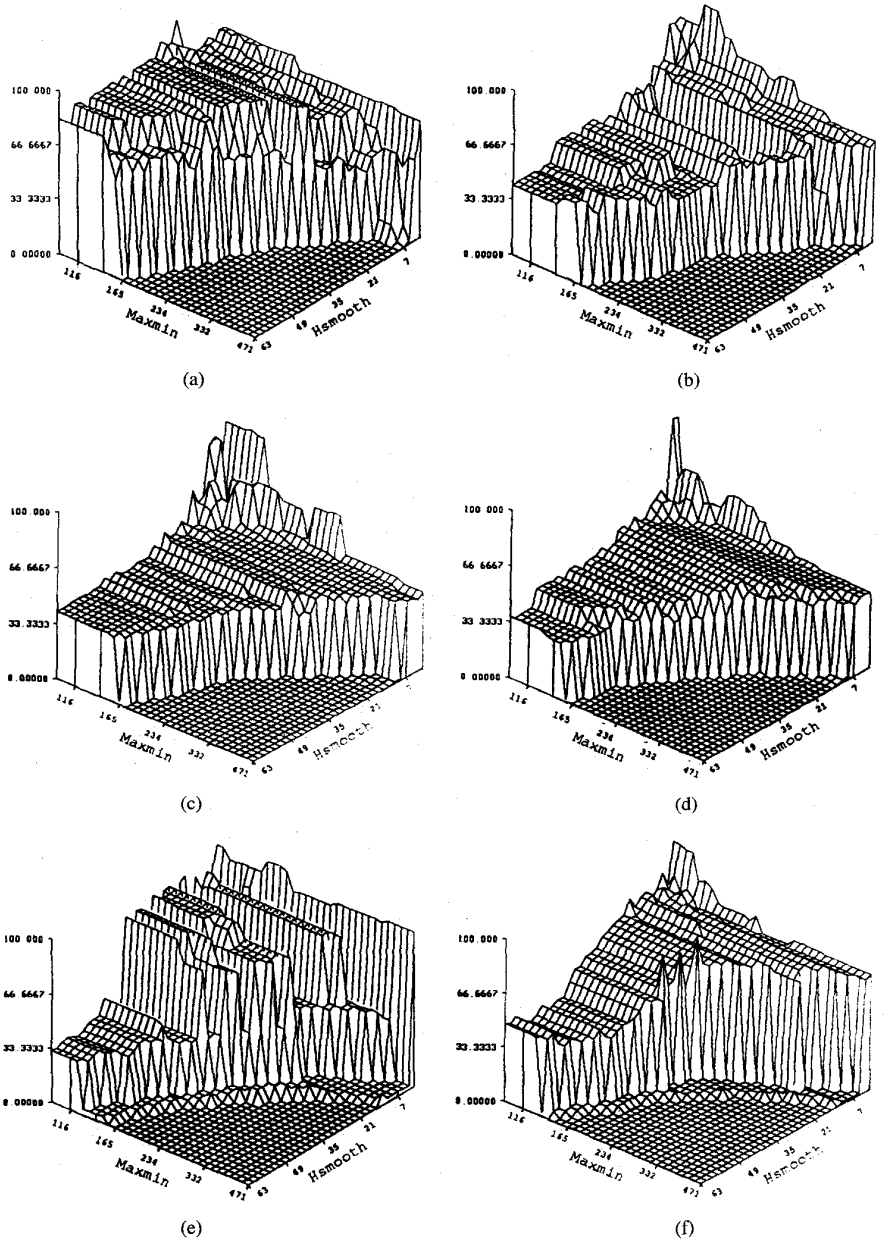


Fig. 13. Combined segmentation quality surfaces for all images in Fig. 9. (a) Frame 1. (b) Frame 6. (c) Frame 10. (d) Frame 11. (e) Frame 13. (f) Frame 20.

(not shown here), we find the distinct trend of the genetic process to shift attention to those areas of the surface with the highest level of fitness. For example, consider the initial search points located on the lower plateau of the Frame 1 surface, i.e., segmentation quality 0.0, in Fig. 14(a). By the final generation, all search points have migrated to the rear corner of the surface where the high quality parameter combinations are located. Similar observations can be made from the Frames 11 and 13 results. By examining the performance charts which indicate the maximum and average fitness values of the population during each generation for each frame, we find that maximum fitness values constantly increase in these charts because the highest fit individual in the population is always retained from one generation to the next. Average fitness, on the other hand, fluctuates as the individuals visit different regions of the surface in search of highly fit areas.

Fig. 15 shows the initial and final segmentation results for Frames 1, 11, and 13. These results are obtained from the individual in the genetic population with maximum fitness, i.e., the best segmentation quality. In these results as well as those for the other frames in the sequence, the lower portion of the car, which is dark because of the car's original color and the lack of highlight or glare from the sun (see Fig. 9), is often merged into the darker background of the foliage

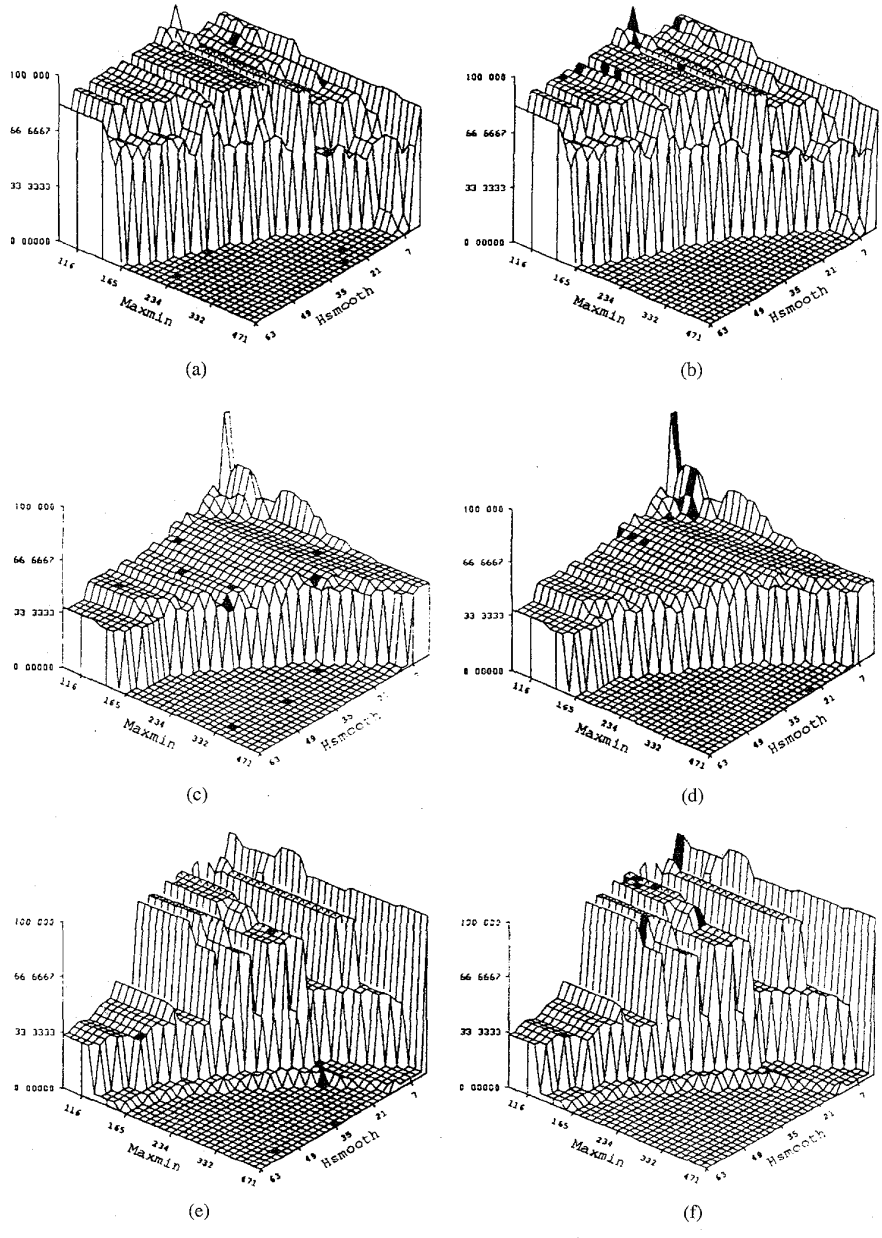


Fig. 14. Starting and final ten search point locations for the training images. Some of the points are not visible in the views shown here. (a) Frame 1 starting locations. (b) Frame 1 final locations. (c) Frame 11 starting locaitons. (d) Frame 11 final locations. (e) Frame 13 starting locations. (f) Frame 13 final locations.

directly behind the car. However, a fairly good silhouette of the upper portion of the car is eventually extracted in the final segmentation results. The characteristics of the image and the limitations inherent in the segmentation algorithm itself are responsible for the behavior described above. An increase in overall segmentation quality between the initial and final results can be seen in each figure as evident from the portion of the car extracted in each image. Sometimes (in Frames 3 and 9, not shown here), the bottom of the car is extracted as a separate region from the background, although this region is still not combined with the top portion of the car to form a single region. Also, the diamond shaped sign to the right of the car is usually extracted in each frame. As the lighting in the frames becomes progressively darker, more and more of the car is combined with the background until in Frame 15, only the very top portion of the car is extracted as a separate region. In this frame, the contrast between the car and the background was so poor, the Phoenix algorithm could not distinguish the borders of the car. Nevertheless, the adaptive image segmentation system was able to optimize the segmentation of the scene within the limits of the Phoenix algorithm.

Once the training phase of the outdoor imagery experiments was complete, the testing phase was begun. The testing phase is designed to measure the reduction in effort obtained by

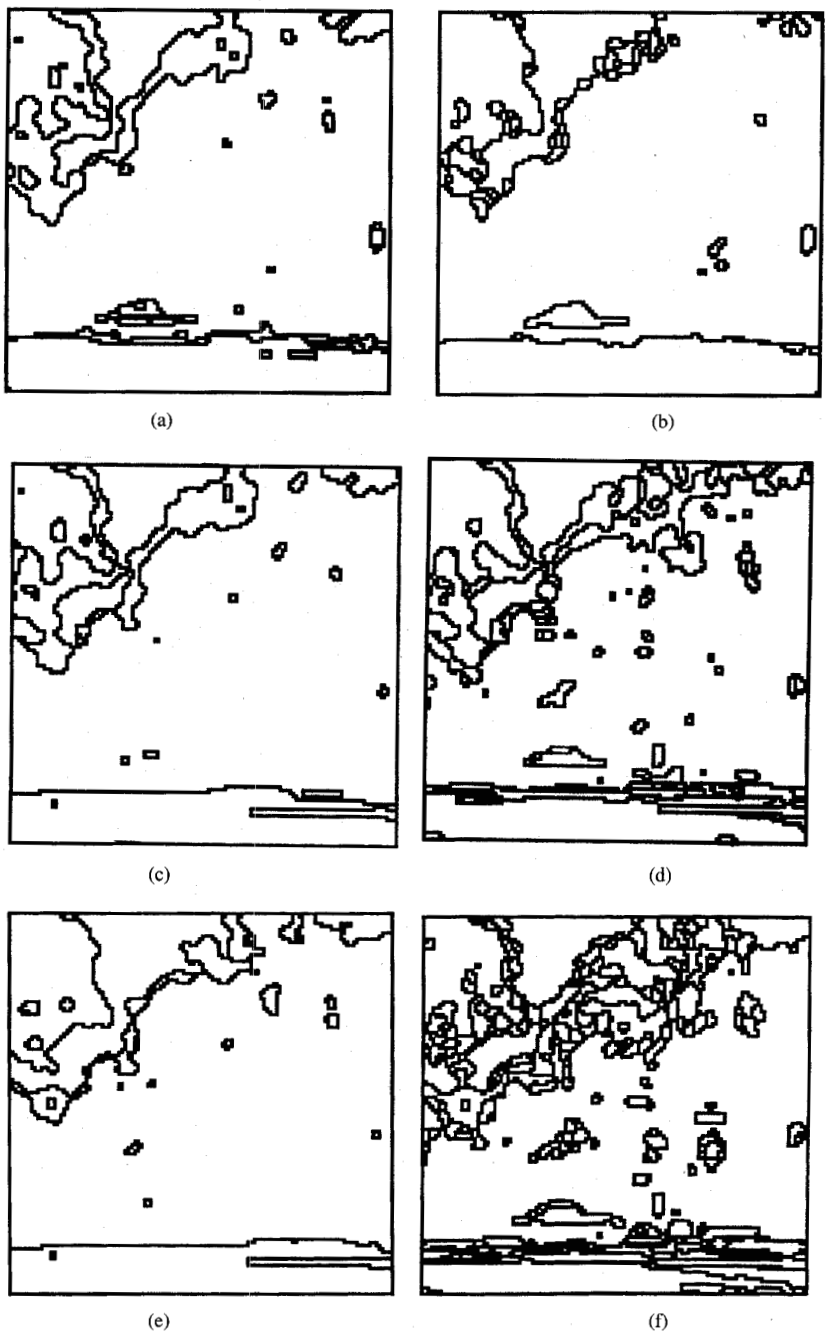


Fig. 15. Segmented images for the outdoor training experiments. (a) Frame 1 initial results. (b) Frame 1 final results. (c) Frame 11 initial results. (d) Frame 11 final results. (e) Frame 13 initial results. (f) Frame 13 final results.

initializing the genetic parameter optimization process with non-random starting points. The final populations from each of the training images $(1, 3, \cdots, 19)$ were combined to create a global population of 100 individuals. From this population, the 10 initial members of each seed population for the testing images $(2, 4, \cdots, 20)$ were selected. The testing was performed in a parallel fashion; the final population for each of the testing images was not placed back into the global population for these tests. The alternative approach to testing, which processes each frame in the outdoor imagery database in a sequential manner and integrates the results into the global population, will be discussed at the end of this section.

Using the seed populations obtained from the global population, the adaptive image segmentation process was invoked on each of the testing images $(2, 4, \cdots, 20)$. The reduction in effort for the genetic process is because of the diversity of the global population. Fig. 16 illustrates the initial seed population and final population for Frames 6, 10, and 20 in the outdoor testing experiments. These figures clearly indicate the high quality of both the initial and final populations for each image.

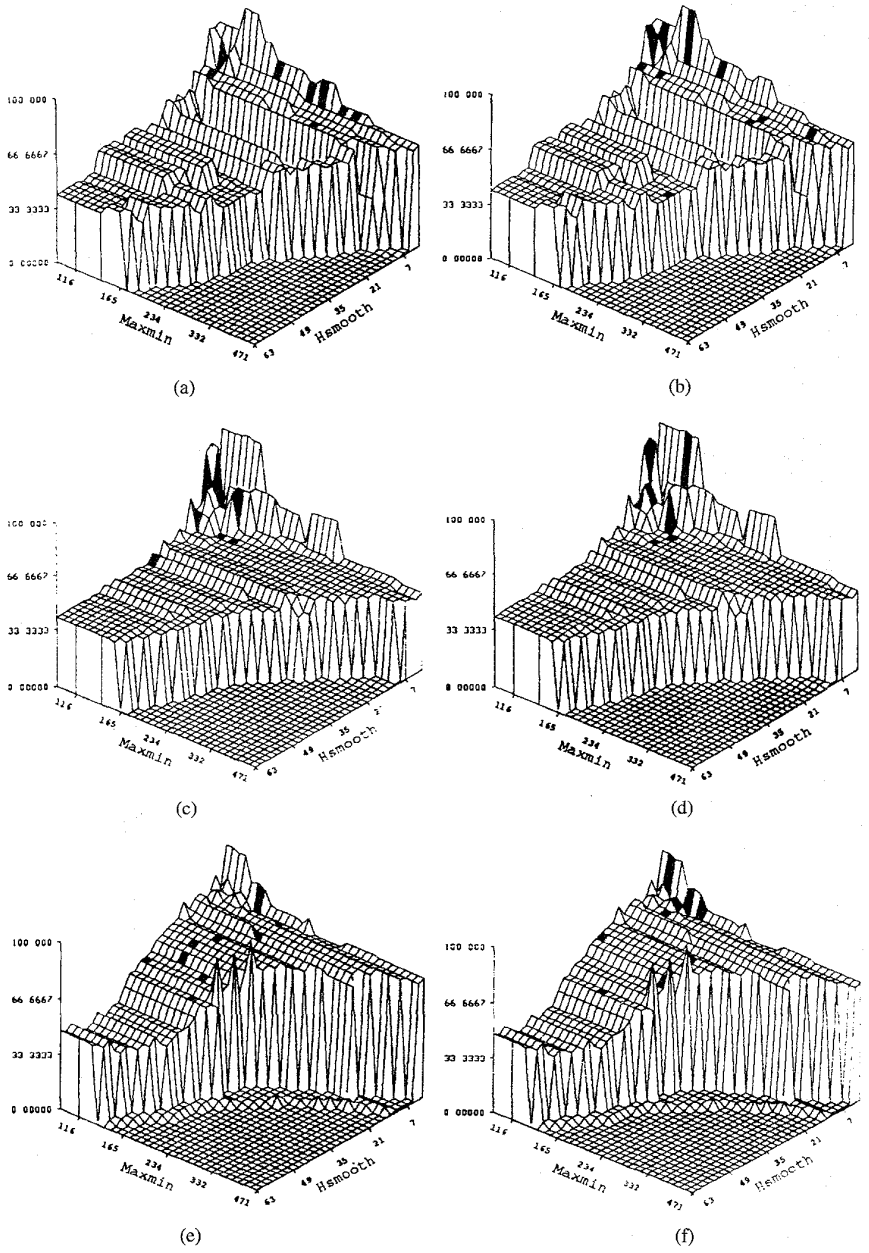


Fig. 16. Starting and final search point locations for the outdoor testing images. (a) Frame 6 starting locations. (b) Frame 6 final locations. (c) Frame 10 starting locations. (d) Frame 10 final locations. (e) Frame 20 starting location. (f) Frame 20 final location.

In every initial population (except for Frames 14 and 16), all points are concentrated on the upper levels of the segmentation quality surface. The single point located on the lower plateau in Frames 14 and 16 are located very near the steep "cliffs" which transition from the low quality segmentation parameter sets (quality $< 10\%$) to the higher quality parameter combinations (quality $> 33\%$). These points are quickly shifted to higher portions of the surface during the early generations of the genetic process. In the final population for each image, all individuals reside in the upper portions of the surface.

The initial and final segmentation results for Frames 6, 10, and 20 are shown in Fig. 17. The improved quality of the initial segmentation results during testing can be visually compared with the initial results acquired during training (Fig. 15). Results for adjacent images, e.g., Frames 10 and 11, in the image sequence can also be compared since they are similar in overall quality. For example, the initial representation of the car region in Frame 10 (Fig. 17(c)) is much better than the initial car region in Frame 11 (Fig. 15(c)). Overall, the experiments reveal the general trend towards improved initial accuracy during the testing stage. The final segmentation results for each of the testing images are also very similar in quality to the training results. Considering that the average number of generations was reduced from 9 during training to 3 during testing,

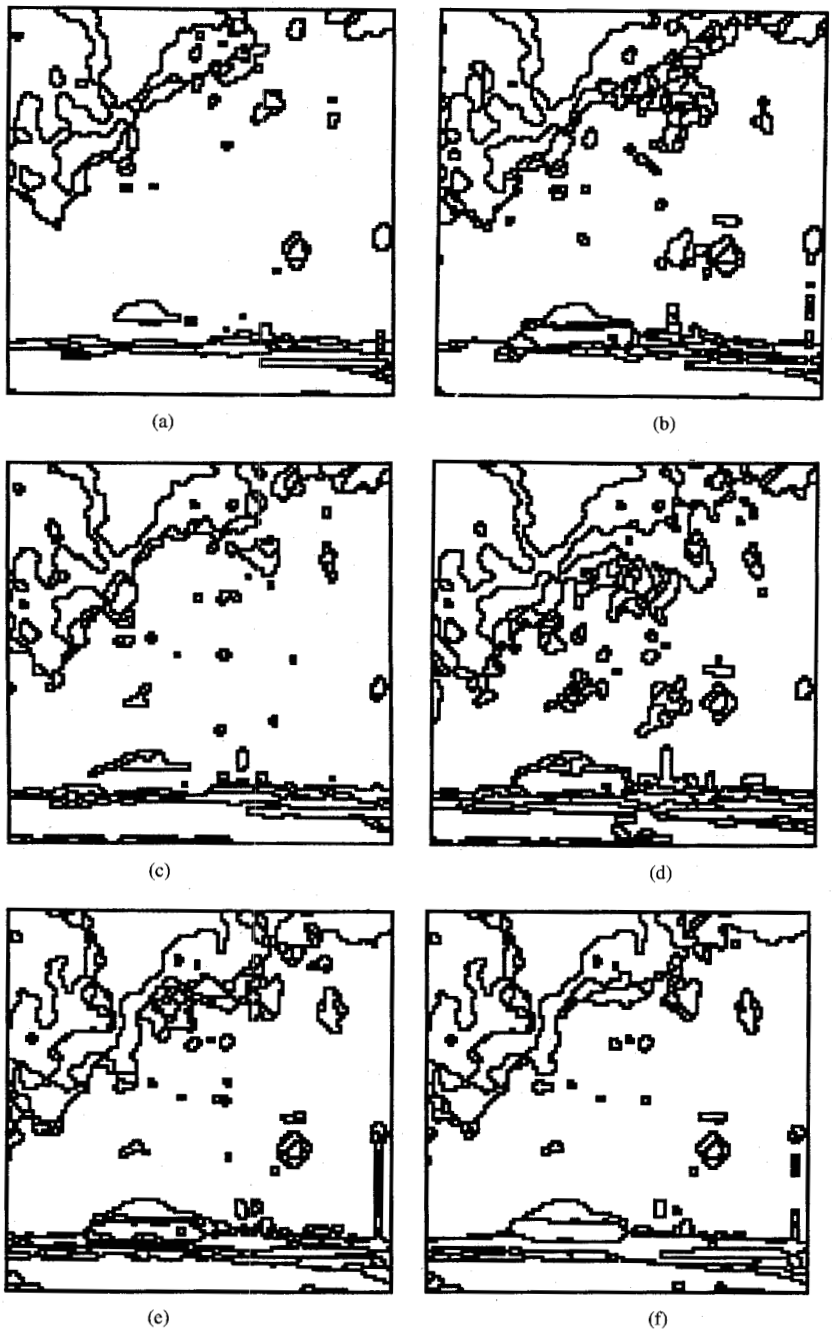


Fig. 17. Segmented images for the outdoor testing experiments. (a) Frame 6 initial results. (b) Frame 6 final results. (c) Frame 10 initial results. (d) Frame 10 final results. (e) Frame 20 initial results. (f) Frame 20 final results.

equivalent segmentation performance during testing represents considerable improvement in the adaptive system's efficiency. On the average, the adaptive segmentation system visits approximately 2.5% of the search space (i.e., ~ 2.5 generations) for the experiments described here for outdoor images.

Since there are no other known adaptive segmentation techniques with a learning capability in both the computer vision and neural networks fields to compare our system with, we measured the performance of the adaptive image segmentation system relative to the set of default Phoenix segmentation parameters [19], [26] and a traditional optimization approach. The *default parameters* have been suggested after extensive amounts of testing by various researchers who developed the Phoenix algorithm [19]. The parameters for the traditional approach are obtained by manually optimizing the segmentation algorithm on the first image in the database and then utilizing that parameter set for the remainder of the experiments. This approach to segmentation quality optimization is currently a standard practice in state-of-the-art computer vision systems.

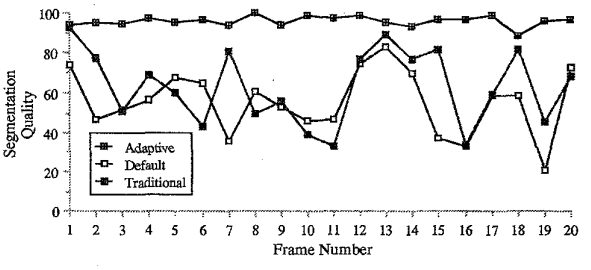


Fig. 18. Comparison of the adaptive image segmentation system with default *Phoenix* performance and the traditional image segmentation approach. The adaptive technique achieves 95.8% segmentation accuracy while the default parameters and the traditional approach obtained only 55.6 and 63.2% accuracies, respectively.

Fig. 18 presents the comparison between our adaptive segmentation system, the results from the default parameter set, and the traditional approach. The average segmentation quality for the adaptive segmentation technique was 95.8%. In contrast, the performance of the default parameters was only 55.6% while the traditional approach provided 63.2% accuracy. As the figure shows, the performance of both of these alternative approaches was highly erratic throughout the sequence of outdoor images. Fig. 19 illustrates the quality of the segmentation results for Frames 1 and 11 using the default parameters and the traditional approach and contrasts this performance with our adaptive segmentation technique. Each result corresponds to the average segmentation performance produced by each technique for the first frame in the outdoor image database. By comparing the extracted car region in each of these images, as well as the overall segmentation of the entire image, it is clear that the adaptive segmentation results are superior to the other methods. Also note that for Frame 1 using the traditional approach, the segmentation quality is initially 95% (Frame 1), which is close to the adaptive segmentation quality. This value indicates that our segmentation evaluation measures are providing information similar to human perceptual performance.

The outdoor experiments described above were conducted in a parallel fashion, i.e., all training and all testing was performed without the aid of previous segmentation experience. Although the testing experiments used the knowledge acquired during training, the tests were still performed in parallel. None of the segmentation experience obtained during testing was applied to subsequent testing images.

B. Comparison of the Adaptive System with Random Search

Several tests were performed to compare the optimization capabilities of the adaptive segmentation system with a simple random walk through the search space. This experiment used only the training images $(1, 3, \cdots, 19)$ from the outdoor image database so that the adaptive system would not benefit from the reuse of segmentation experience from one image to the next. The intent of this restriction was to measure the efficiency of the genetic algorithm in optimizing a complex surface. In addition, the stopping criteria for the adaptive system was simplified so that when a surface point with 95% segmentation quality or better was located, the optimization process would terminate. The random walk algorithm searched the segmentation quality surface by visiting points randomly and used the same 95% stopping criteria. Finally, in order to insure correctness of the results, each segmentation quality surface was optimized by each technique 100 times and the results are averaged to create the performance figures discussed below.

Fig. 20(a) presents a comparison of the efficiency for the two techniques described above. The bars represent the total number of points visited on the surface using each technique for each of the images and the average number of points visited for each approach. As the average values show, the adaptive technique is far superior to the random walk approach. In addition, the average number of points visited by the adaptive approach is $\sim$ 6.9% of the total number of points on the surface, compared to the earlier experiments where we processed $\sim$ 2.5% of the surface, since we have not reused any segmentation experience gained from processing earlier images.

Fig. 20(b) contrasts the segmentation quality achieved by the two techniques. Since the adaptive segmentation technique insures the achievement of a near global maximum for each image, we modified the random walk approach so that it would terminate after the same number of visited locations required by the adaptive technique. The maximum segmentation quality achieved by the random approach was then compared with the adaptive system. On the average, the adaptive system achieved 99.3% segmentation quality after the number of segmentations shown in Fig. 20(a). In comparison, the random walk achieved only 81.4% of the maximum quality for the same number of segmentations for each image.

C. Effectiveness of the Reproduction and Crossover Operators

A number of tests were performed to demonstrate the effectiveness of the reproduction and crossover operators in the adaptive image segmentation system. The optimization capability of the pure genetic algorithm was compared with two variations of the genetic algorithm. The first variation of the pure genetic algorithm was implemented without a reproduction operator. Instead of reproducing individuals according to their fitness values, the algorithm selected the individuals at random for further genetic operator action with the restriction that any individual be selected only once. The second variation of the genetic algorithm simply skipped a crossover operator. To ensure that this approach generates about the same number of offsprings as the pure genetic algorithm, the mutation rate of this approach was increased to the crossover rate (0.8) of the genetic process. The stopping criteria for each technique is to locate a surface point with 95% or higher segmentation quality. In order to ensure correctness of the results each image was tested by each technique 100 times and the results were averaged to create the performance figure. Fig. 21 presents the comparison of the optimization capability for three techniques. As the histograms show, the pure genetic algorithm results are much better than the results of the other two approaches for both the training and testing experiments. This demonstrates that the reproduction and crossover operators are critical for the success of genetic algorithms.

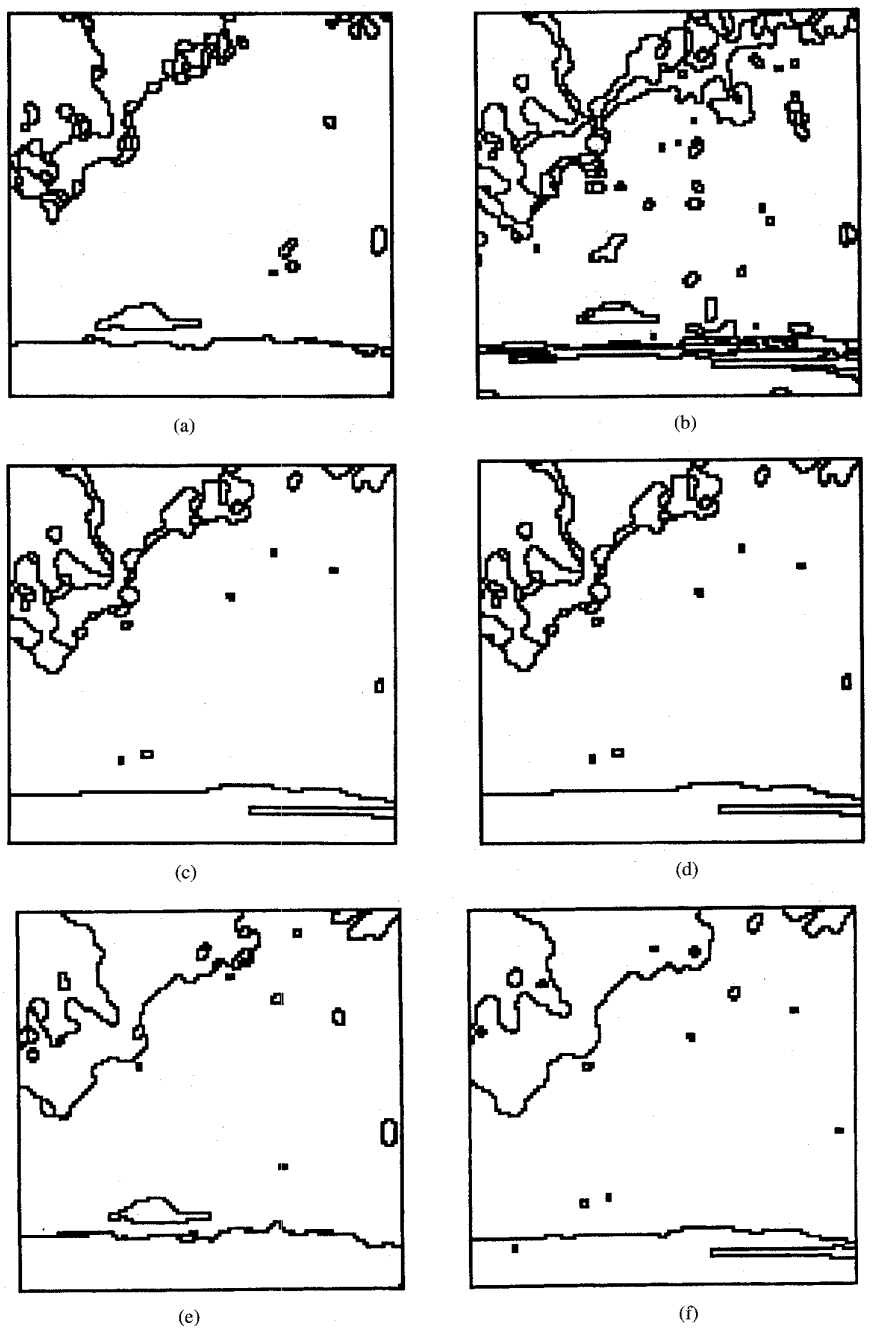


Fig. 19. Segmentation results for the adaptive technique, the default parameters, and the traditional approach using the outdoor images. (a) Frame 1 adapative technique results. (b) Frame 11 adaptive technique results. (c) Frame 1 default parameter results. (d) Frame 11 default parameter results. (e) Frame 1 traditional approach results. (f) Frame 11 traditional approach results.

D. Demonstration of the Learning Behavior

To measure the improvement in efficiency achieved by immediately reusing segmentation experience, we also conducted a set of experiments. These experiments were designed to investigate the reduction in computational effort obtained by processing the images in a sequential rather than parallel manner. All genetic algorithm parameters (population size, crossover rate, etc.) and the evaluation criteria (five separate quality measures, segmentation quality threshold of 95%) were retained from the earlier experiments. Finally, a comparison of the sequential and parallel testing experiments is presented.

1) Sequential testing experiment: Three separate *sequential tests* were performed. In each case, the order of the images presented to the adaptive image segmentation system was modified to determine the sensitivity of the sequential process to variations in the image sequences. The first test processed the outdoor images in their original order, i.e., Frames $1, 2, 3, \cdots, 20$. The second test processed the odd

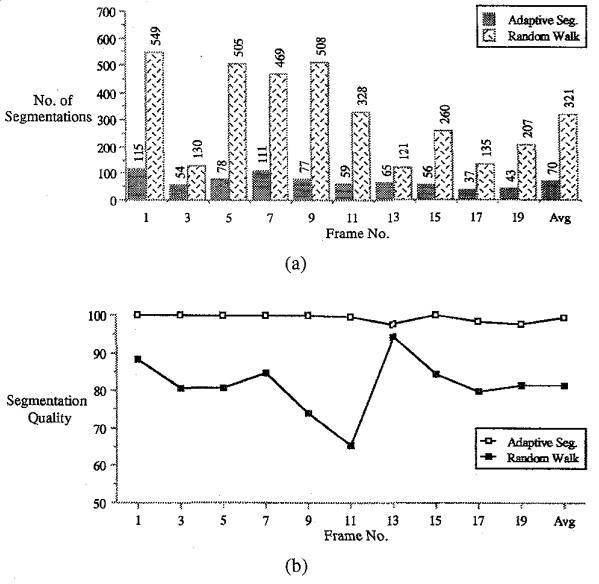


Fig. 20. Performance comparison of the adaptive image segmentation system and the random walk technique. (a) Comparison of the computational efforts for the same segmentation quality. (b) Comparison of the segmentation quality with the same computational effort. The segmentation quality represents the percentage of the maximum quality for each frame of the training images.

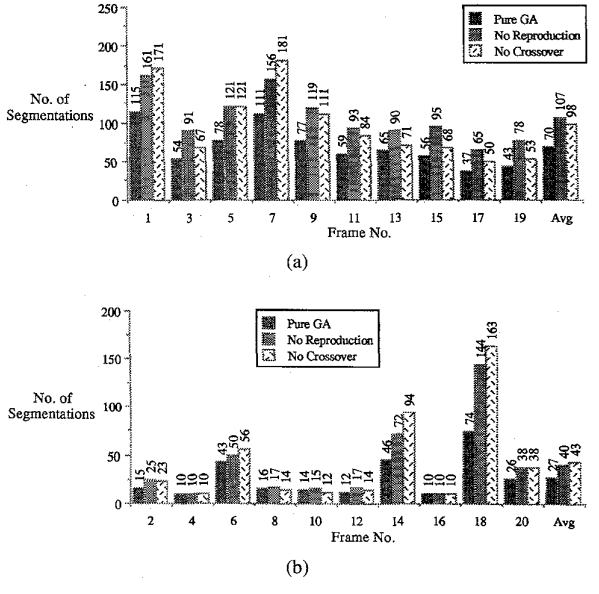


Fig. 21. Performance comparison of the pure genetic algorithm and its two variations. The superior performance of the pure genetic algorithm demonstrates the effectiveness of the reproduction and crossover operators.

numbered images first and then the even numbered images, i.e., Frames $1, 3, 5, \cdots, 19$ followed by Frames $2, 4, \cdots, 20$. This order was chosen so that we could compare the performance of the sequential processing with the parallel experiments performed earlier. Finally, the third test altered the sequence of images to simulate a multi-day scenario where the frequency of image collection decreases to approximately one hour. The order of the images in this test is $\{1, 5, 9, 12,$

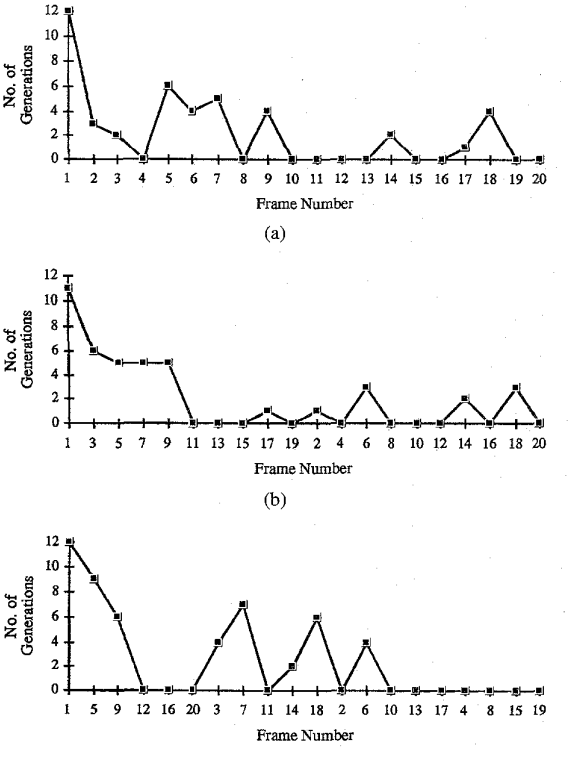


Fig. 22. Performance of the adaptive image segmentation system for the sequential experiments. (a) Single day test results. (b) Double day test results. (c) Multiple day test results.

$16, 20, 3, 7, 11, 14, 18, 2, 6, 10, 13, 17, 4, 8, 15, 19\}$. Each group of images in the sequence of Frames $\{1, 5, 9, 12, 16, 20\}, \{3, 7, 11, 14, 18\}, \{2, 6, 10, 13, 17\}$, or $\{4, 8, 15, 19\}$) was designed to represent a collection of images acquired on a different day. Thus, using the sequence of images described above, we have simulated a four day long collection of images.

For each of the three tests, the genetic population of the first frame in the image sequence was randomly selected. Once the segmentation performance for that frame was optimized by the genetic algorithm, the final population from that image was used to create the initial global population. This global population was then used to select the seed population for subsequent frames in the image sequence. The global population size was set to 100 for these experiments to insure a diversity of segmentation experience in the population. While the size of the global population remained below 100 members (prior to processing 10 frames), the final populations for each image were merely added to the current global population. After the size of the global population reached 100 individuals, the final populations from each successive image had to compete with the current members of the global population. This competition was based on the fitness of the individuals; highly fit members of a new local population replaced less fit members of the global population, thus keeping the size of the global population constant. Fig. 22 presents the performance results achieved by the adaptive image segmentation system during each of the three sequential tests.

Single Day Sequential Test—Fig. 22(a) illustrates the performance of the system for the single day sequence (*first test*). The number of generations for the first frame is quite large since we started from a random collection of search points. The experience gained in processing the first frame is immediately utilized during the second frame. The number of generations has been reduced from 12 to 3. Similarly, for Frames 3 and 4, the number of generations decreases each time. Although the number of generations does increase at several points beyond the fourth frame, the overall trend of this plot does indicate a reduction in computational effort. This claim is evident by noting that for the 20 frames of outdoor imagery in this sequence, the adaptive image segmentation system optimizes the segmentation quality of 50% (10 out of 20) of these images using the information present in the global population. No iterations of the genetic generations are necessary in these cases.

Odd-Even Image Sequence Test—Fig. 22(b) provides similar evidence of learning and computational savings for the sequence of images used in the *second test*. Note that the initial slope of the graph in this figure is not as steep as in Fig. 22(a). This difference is due to the fact that the image intervals have increased in this experiment (e.g., we take every other image instead of every image). Thus, the knowledge previously acquired by the adaptive process is not as immediately relevant to subsequent images as it was during the first test. However, once we have processed all odd numbered images, the number of generations required during the even numbered images is substantially smaller. It is interesting to note that the even numbered images which require several generations (Frames 6, 14, and 18) in this test also required similar efforts in the first test (Fig. 22(a)). This correlation implies that the knowledge currently in the global population was not sufficient to optimize the segmentation quality of these images without some assistance from the genetic algorithm. Finally, note that as was the case in the first test, the adaptive image segmentation system optimizes the segmentation quality of half the image sequence (10 of 20 frames) without invoking the genetic process.

Multiple Day Sequential Test—Fig. 22(c) presents the computational efforts required for the multi-day simulation in the *third test*. Once again, we can see the difference in the initial slope of the graph, which is due to the order in which the images are encountered. In this case, since there is an even wider separation between the images than in the two previous tests, the number of generations required for the first few frames is much higher. Additionally, with the exception of some local irregularities, the graph in Fig. 22(c) shows the cyclical nature of the multi-day process. The irregularities are attributed to the troublesome frames (6, 14, and 18) described earlier. The images in the first "day" (frames $\{1, 5, 9, 12, 16, 20\}$) show a continually decreasing level of computational effort. When the second sequence (frames $\{3, 7, 11, 14, 18\}$) is encountered, the effort increases temporarily as the adaptive process fills in the knowledge gaps present as a result of the differences between the images in each sequence. The image sequence for the third "day" (frames $\{2, 6, 10, 13, 17\}$) was handled with almost no effort by the genetic learning. Finally, the fourth

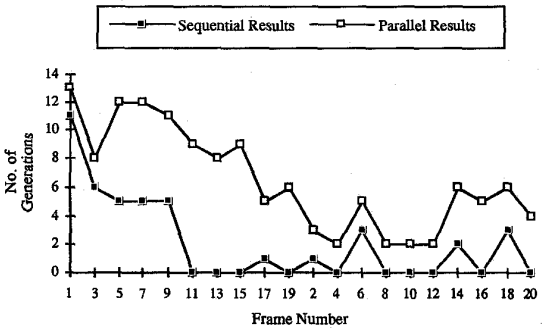


Fig. 23. Comparison of the sequential and parallel experiments. In every frame, the sequential experimental results outperform the parallel experimental results.

image sequence (frames $\{4, 8, 15, 19\}$) requires no effort by the genetic learning at all; each image is optimized by the information stored in the global population. Note that the third test contains the largest number of frames processed with no help from the genetic algorithm. Twelve of the twenty frames in this test were optimized using the global population.

2) Sequential results versus parallel results: A final comparison contrasts the performance of the sequential experiments and the parallel experiments described earlier. Fig. 23 examines the reduction in effort obtained by the sequential processing tests. The performance figures for the parallel results are obtained from the data obtained during training experiments. For each image in the outdoor database, the sequential tests provide fewer numbers of generations in order to optimize the segmentation quality. As before, we see that Frames 6, 14, and 18 required additional processing effort regardless of the approach used during the experiments. The results in Fig. 23 provide strong evidence for the utilization of a sequential approach to the image segmentation optimization problem. The above tests also demonstrate that the process of adaptive image segmentation can be performed in a completely unsupervised mode.

VI. Conclusions

We have shown the ability of the adaptive image segmentation system to provide high quality ($> 95\%$) segmentation results in a minimal number of segmentation cycles. The performance improvement provided by the adaptive system was consistently greater than $\sim 30\%$ over the traditional approach or the default segmentation parameters [19], [26]. Recently we have carried out experiments where we have used *hsmooth, maxmin, splitmin,* and *height* parameters of the Phoenix algorithm to achieve adaptive image segmentation. The size of search space in these experiments is over 1 million and the number of points that are visited on the surface varies from 0.3–0.6% for 95% quality of segmentation for outdoor images [4]. The genetic parameters used for scaling experiments consist of four crossover points, crossover rate of 0.8, mutation rate of 0.01, and short-term population size of 10. Note that, as the number of segmentation parameters for adaptation increases, the number of points to be visited

on the surface will also increase. Therefore, the computational requirements change with the number of parameters. However, genetic algorithms are not subject to the "curse of dimensionality," much like Hough transform techniques. Unlike Hough transform, which is essentially an exhaustive search technique, it is expected that the genetic algorithm will visit only a small percentage of the search space to achieve the global maximum.

Further research efforts are planned to extend the scope of the current adaptive system. We plan to use a data set with dramatic environmental variations and we will utilize several segmentation algorithms and a larger number of segmentation parameters. Ultimately, we will incorporate the adaptive segmentation component into our complete vision system.

REFERENCES

[1] D. H. Ackley, "Stochastic iterated genetic hillclimbing," Ph.D. thesis CMU-CS-87-107, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Mar. 1987.

[2] B. Bhanu, "Automatic target recognition: State of the art survey," *IEEE Trans. Aerosp. Electron Syst.*, vol. AES-22, no. 4, pp. 364–379, July 1986.

[3] ——, "Image understanding research at Honeywell," in *Proc. DARPA Image Understanding Workshop,* Sept. 1990, pp. 70–75.

[4] B. Bhanu, R. N. Braithwaite, W. Burger, S. Das, S. Rong, and X. Wu, "Multistrategy learning for image understanding," Technical Report to AFOSR/ARPA, Feb. 1994.

[5] B. Bhanu, S. Lee, and S. Das, "Adaptive image segmentaton using multi-objective evaluation and hybrid search methods," *AAAI Fall Symp. Machine Learning in Computer Vision,* Oct. 1993.

[6] B. Bhanu, S. Lee, and J. C. Ming, "Self-optimizing image segmentaiton system using a genetic algorithm," in *Proc. Fourth Int. Conf. on Genetic Algorithms,* San Diego, CA, July 1991, pp. 362–369.

[7] L. Davis and M. Sttenstrup, "Genetic algorithms and simulated annealing: An overview," in *Genetic Algorithms and Simulated Annealing,* L. Davis, Ed., 1987.

[8] K. DeJong, "Learning with genetic algorithms. An overview," *Machine Learning,* vol. 3, pp. 121–138, 1988.

[9] J. M. Fitzpatrick and J. J. Grefenstette, "Genetic algorithms in noisy environments," *Machine Learning,* vol. 3, pp. 101–120, 1988.

[10] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition,* vol. 13, pp. 3–16, 1981.

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, 1984.

[12] A. M. Gillies, "Machine learning procedures for generating image domain feature detectors," Ph.D. thesis, Dept. of Computer and Communication Sciences, University of Michigan, Ann Arbor, Apr. 1985.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA. Addison-Wesley, 1989.

[14] J. J. Grefensette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst. Man Cyber.*, vol. SMC-16, no. 1, pp. 122–128, Jan. 1986.

[15] R. M. Haralick and L. G Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing,* vol. 29, pp. 100–132, 1985.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, 1975.

[17] ——, "Escaping brittlenes: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems," in *Machine Learning: An Artificial Intelligence Approach. Vol. II,* R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds.. New York: Morgan Kaufmann, Inc., 1986, pp. 593–623.

[18] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science,* vol. 220, no. 4598, pp. 671–680, May 13, 1983.

[19] K. I. Laws, "The Phoenix image segmentation system: Description and evaluation," SRI International Technical Note No. 289, Dec. 1982.

[20] V. R. Mandava, J. M. Fitzpatrick, and D. R Pickens, III, "Adaptive search space scaling in digital image registration," *IEEE Trans. Med. Imag.*, vol. 8, no. 3, pp. 251–262, Sept. 1989.

[21] T. Matsuyama, "Expert systems for image processing: Knowledge-based composition of image analysis processes," *Computer Vision, Graphics, and Image Processing,* vol. 48, pp. 22–49, 1989.

[22] D. L. Milgram, "Region extraction using convergent series," *Computer Graphics and Image Processing,* vol. 11, pp. 1–12, 1979.

[23] J. Ming and B. Bhanu, "A multistrategy learning approach for target model recognition, acquisition and refinement," in *Proc. DARPA Image Understanding Workshop,* Sept. 1990, pp. 742–756.

[24] R. Ohlander, K. Price, and D. R. Reddy, "Picture segmentation using a recursive region splitting method," *Computer Graphics and Image Processing,* vol. 8, pp. 313–333, 1978.

[25] P. G. Selfridge, "Reasoning about success and failure in aerial image understanding," Ph.D. thesis, Department of Computer Science, 1982.

[26] S. Shafer and T. Kanade, "Recursive region segmentation by analysis of histograms," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* 1982, pp. 1166–1171.

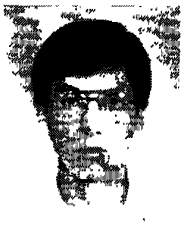
[27] E. D. Sontag and H. J. Sussmann, "Image restoration and segmentation using the annealing algorithm," in *Proc. 24th Conf. on Decision and Control,* Dec. 1985, pp. 768–773

[28] J. S. Weszka, "A survey of threshold selection techniques," *Computer Graphics and Image Processing,* vol. 7, pp. 259–265, 1978.

Bir Bhanu (S'72–M'82–SM'87) received the B S. degree (with Honors) in electronics engineering from the Institute of Technology, BHU, Varanasi, India, the M.E degree (with Distinction) in electronics engineering from the Birla Institute of Technology & Science, Pilani, India, the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, the Ph.D. degree in electrical engineering from the Image Processing Institute, University of Southern California, Los Angeles, and the M.B.A. degree from the University of California, Irvine.

Since 1991, Dr. Bhanu has been a Professor of Electrical Engineering and Computer Science and Director of Visualization and Intelligent Systems Laboratory at the University of California, Riverside. Prior to that he was a Senior Honeywell Fellow at Honeywell Systems & Research Center, Minneapolis, MN. He has also worked with the University of Utah, IBM San Jose Research Laboratory, INRIA-France and Ford Aerospace & Communications Corporation.

Dr. Bhanu has been the principal investigator of various programs from ARPA, NASA, AFOSR, ARO, NSF and other agencies and industries. Currently he is the principal investigator on grants from ARPA and others in the areas of learning and vision, target recognition, and machine vision applications. He has five U.S. and international patents in Computer Vision and over 150 reviewed technical publications in the areas of computer vision, image processing, robotics and artificial intelligence. He was the guest editor or co-editor of a special issue of IEEE COMPUTER on "CAD-Based Robot Vision," the special issue of *Journal of Robotic Systems* on "Passive Ranging for Robotic Systems," and special issues of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE on "Learning in Computer Vision," IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION on "Perception-Based Real-World Navigation," and *International Journal of Machine Vision and Application* on "Innovative Applications of Computer Vision," and an upcoming special issue of IEEE TRANSACTIONS ON IMAGE PROCESSING. He is on the editorial board of the *Journal of Mathematical Imaging and Vision*, the *Journal of Pattern Recognition*, and the *International Journal of Machine Vision and Applications*. He is the co-author of books on "Computational Learning for Adaptive Computer Vision," (Plenum, forthcoming), "Genetic Learning for Adaptive Image Segmentation" (Kluwer, 1994). and "Qualitative Motion Understanding" (Kluwer, 1992). He was the general chair for the first IEEE Workshop on Applications of Computer Vision (Palm Springs, CA, 1992) and he is the General Chair for the IEEE Conference on Computer Vision and Pattern Recognition (San Francisco, CA, 1996). He is a member of ACM, AAAI, Sigma Xi, Pattern Recognition Society, and SPIE.



Sungkee Lee (M'94) received the B.S. and M.S. degrees in electrical engineering from Seoul National University in South Korea in 1979 and 1981, respectively. He received the Ph.D. degree in computer science from the University of Utah, Salt Lake City, in 1992.

Currently, Dr. Lee is an Assistant Professor of Computer Science at Kyungpook National University, South Korea. His interests are computer vision, machine learning, neural networks, fuzzy systems and scientific visualization.

John Ming worked as a research scientist with the Signal and Image Processing group, Honeywell's Systems and Research Center, from 1987 to 1991. His research efforts focused on computer vision and machine learning for satellite photointerpretation applications under DARPA/ORD sponsorship.

Since 1991, he has led the Image Understanding department, AT&T Human Interface Technology Center, Atlanta, GA. His current interests are commercial and industrial machine vision systems. Recent focus has centered on document imaging applications and real-time image processing systems for retail/financial environments.