Adaptive Image Segmentation Using Genetic and Hybrid Search Methods

BIR BHANU, Senior Member, IEEE
University of California at Riverside

SUNGKEE LEE
Kyungpook National University
South Korea

SUBHODEV DAS, Member, IEEE
University of California at Riverside

This paper describes an adaptive approach for the important image processing problem of image segmentation that relies on learning from experience to adapt and improve the segmentation performance. The adaptive image segmentation system incorporates a feedback loop consisting of a machine learning subsystem, an image segmentation algorithm, and an evaluation component which determines segmentation quality. The machine learning component is based on genetic adaptation and uses (separately) a pure genetic algorithm (GA) and a hybrid of GA and hill climbing (HC). When the learning subsystem is based on pure genetics, the corresponding evaluation component is based on a vector of evaluation criteria. For the hybrid case, the system employs a scalar evaluation measure which is a weighted combination of the different criteria. Experimental results for pure genetic and hybrid search methods are presented using a representative database of outdoor TV imagery. The multiobjective optimization demonstrates the ability of the adaptive image segmentation system to provide high quality segmentation results in a minimal number of generations. The results of the hybrid method show the performance improvement over the pure GA.

Manuscript received August 13, 1993; revised July 26, 1994.

IEEE Log No. T-AES/31/4/14138.

This work was partly supported by ARPA/AFOSR Grant F49620-93-1-0624. The contents of the information does not necessarily reflect the position or the policy of the Government.

Authors' current addresses: B. Bhanu, College of Engineering, University of California, Riverside, CA 92521-0425; S. Lee, Kungpook National University, South Korea; S. Das, Princeton Electronic Board Co., Princeton, NJ.

I. INTRODUCTION

Image segmentation is an important and, perhaps, the most difficult low-level task. Segmentation refers to the grouping of image elements that exhibit "similar" characteristics. All subsequent interpretation tasks—feature extraction, object recognition, and classification—rely heavily on the quality of the segmentation process. The difficulty arises when the segmentation performance needs to be adapted to the changes in image quality. Image quality is affected by variations in environmental conditions, imaging devices, time of day, etc. Despite the large number of segmentation techniques presently available [4, 11, 14], no general methods have been found that perform adequately across a diverse set of imagery. When presented with a new image, selecting the appropriate set of algorithm parameters is the key to effectively segmenting the image [5]. However, no segmentation algorithm can automatically generate an "ideal" segmentation result in one pass (or in an open-loop manner) over a range of scenarios encountered in real-world applications. Any technique, no matter how "sophisticated" it may be, will eventually yield poor performance if it cannot adapt to the variations in unstructured scenes.

In reality, there exist several factors which make the parameter adaptation process very difficult. *First*, the number of parameters present in a typical segmentation algorithm is usually quite large. Therefore, the search for the optimal parameter set can be prohibitively large, unless the parameter space is traversed in a highly efficient manner. *Second*, the parameters mutually interact in a complex, nonlinear fashion, which makes it difficult or impossible to model their behavior in an algorithmic or rule-based fashion. *Third*, since variations between images cause changes in the segmentation results, the objective function that represents segmentation quality also varies from image to image. Consequently, 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 is no single, universally accepted measure of segmentation performance available with which to uniquely define the quality of the segmented image.

Consequently, there exists a need to apply an adaptive segmentation 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. While there are adaptive threshold selection techniques [18, 22, 25] for segmentation, these techniques do not accomplish any learning from experience to improve the performance of the system over time. 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.

This work describes an adaptive image segmentation technique that uses a GA and a GA plus hill climbing (HC) as the machine learning components. The key elements of the adaptive image segmentation system are as follows:

1) A closed-loop feedback control technique which provides an adaptive capability. The feedback loop consists of a learning component, an image segmentation algorithm, and a segmented image evaluation component.

2) A learning subsystem which optimizes segmentation performance on each individual image and accumulates segmentation experience over time to reduce the effort needed to optimize subsequent images.

3) 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.

4) 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 are optimized simultaneously (using a GA) and in a weighted combination (using a GA plus HC). When the local and global measures are to be optimized simultaneously, the problem becomes one of *multiobjective* optimization.

5) The learning subsystem is very fundamental in nature and is not dependent on any specific segmentation algorithm or type of sensor data (visible, infrared, laser, etc). The performance of the overall adaptive system is limited by the capabilities of the segmentation algorithm, but the results are optimal for a given image based on the evaluation criteria.

The next section discusses image segmentation as an optimization problem. Consequently, it argues that the GA is an appropriate search technique for the parameter adaptation task. It also explores the combination of the GA with other search techniques, such as HC, which draws on the mutual strengths of the individual techniques to result in an efficient hybrid algorithm. A brief introduction to the problem of multiobjective optimization is given Section III. 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 using both the genetic and the hybrid algorithms. Finally, Section VI provides the conclusions of this paper.

II. IMAGE SEGMENTATION AS AN OPTIMIZATION PROBLEM

In the absence of any rigorous theory, the problem of image segmentation is best described in terms of its goal. The criteria for good segmentation are [14], 1) the segmented regions should be uniform and homogeneous with respect to some characteristic, such as gray value or texture, 2) region interiors should be free of holes and region boundaries should be smooth and spatially accurate, and 3) adjacent regions should be differing significantly based on the characteristic on which they are uniform. If one represents this criteria set in terms of a hypothetical function, then the problem of (good) segmentation is one of optimizing this objective function by selecting appropriate segmentation parameters. Fig. 1(a) illustrates an objective function that is typical of an image segmentation process. The figure depicts an application in which only two segmentation parameters are being varied, as indicated by the x and y axes. The z axis indicates 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. As an example, Figs. 1(b)–(d) show an indoor image and the corresponding global and local segmentation quality (to be discussed in Section IV D) surfaces and Figs. 1(e)–(f) show the segmentation results corresponding to the global and local segmentation evaluation criteria.

A. Selection of Optimization Technique

The multimodality of the segmentation criterion function exemplified in Fig. 1 emphasizes the need for a highly effective search strategy 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 [2, 13, 26] are as follows:

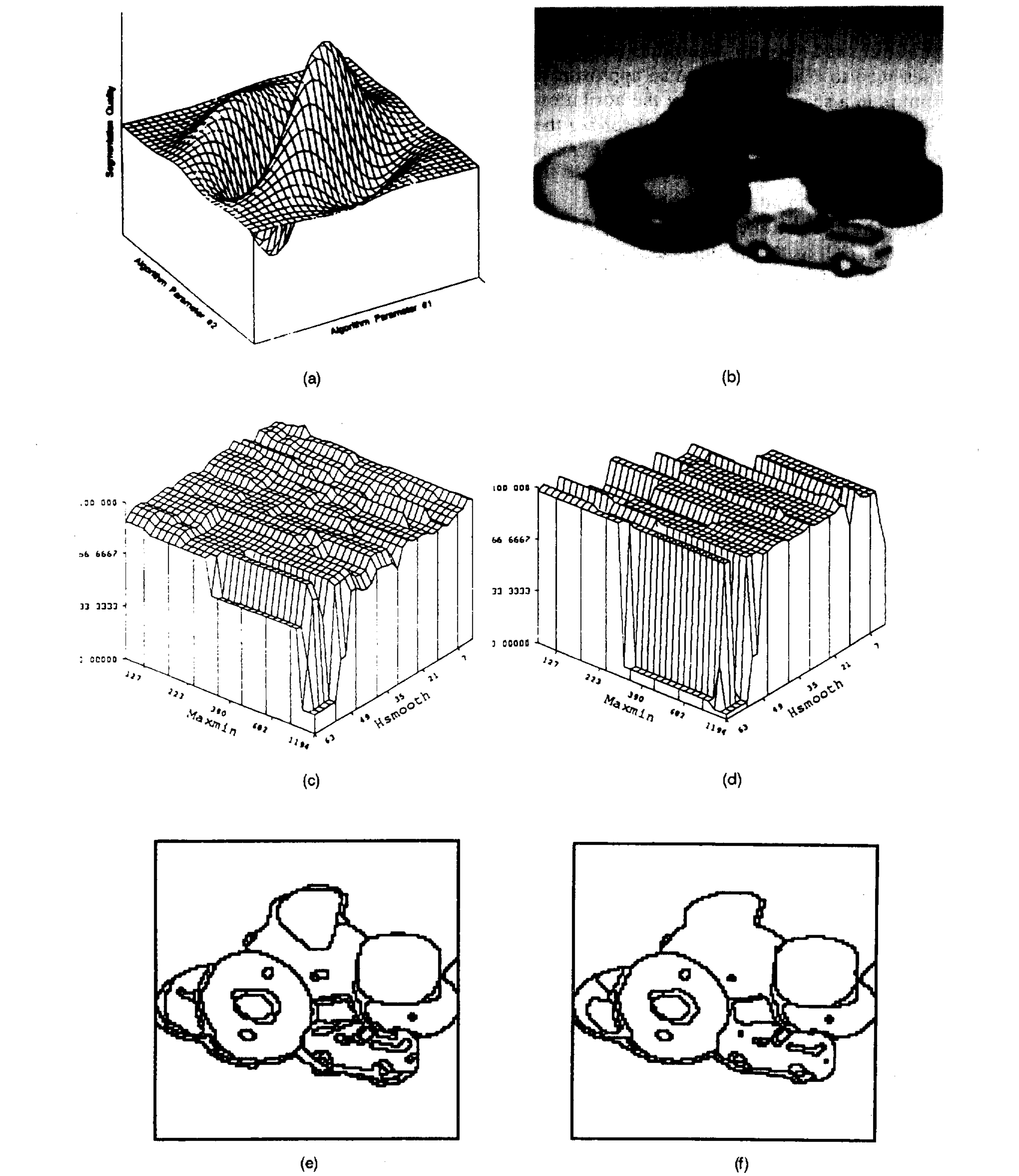


Fig. 1. Example of adaptive image segmentation task. (a) Typical objective function which must be optimized in adaptive image segmentation problem. (b) Indoor image for segmentation. (c) Global segmentation quality surface. (d) Local segmentation quality surface. (e) Global segmentation results. (f) Local segmentation results.

1) *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;

2) *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.

3) *Partial Knowledge Techniques* (*HC, beam search, best first, branch and bound, dynamic programming,* A^*): HC 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*.

4) *Knowledge-Based Techniques (production rule systems, heuristic methods)*: These systems have a limited domain of rule applicability, tend to be *brittle* [16], and are usually difficult to formulate. Further, the visual knowledge required by these systems may not be representable in knowledge-based formats.

GAs are a family of adaptive search methods that are modeled after genetic evolution process. The search process is independent of the problem domain. The basic elements of a GA are called *knowledge structures* or *individuals*. A collection of individuals is referred to as a *population*. At each iteration, known as a *generation*, each individual is reproduced and recombined with others on the basis of its fitness. The learning operators that are responsible for creating a new generation of individuals consist of a "mating operator" (it selects individuals according to their fitness values and produces offsprings through the process of *reproduction*) and "genetic operators" (these determine the genetic makeup of offspring from the genetic material of the parents through the processes of *crossover* and *mutation*). The expected number of times an individual is selected for recombination is proportional to its fitness relative to the rest of the population. The mechanics of reproduction and crossover are fairly simple, involving nothing more complex than random number generation, string copying, and some partial string exchanges. Nevertheless, analytical and empirical studies have demonstrated that the combined emphasis of reproduction and crossover gives GAs much of their power [13]. Mutation provides for occasional disturbances in the crossover operation by inverting one or more genetic elements during reproduction. An abstract procedure of a simple GA 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;
```

The inherent power of GAs lies in their ability to exploit accumulating information about an initially unknown domain in a highly efficient manner. By allocating more reproductive occurrences to above average individuals, GAs can bias subsequent search towards the more productive subspaces containing groups of highly fit individuals. The bias is attributed to certain important similarities that exist among these highly fit individuals. Holland [15] introduces the framework of schemata as a means of understanding the interaction among the various genetic individuals or strings. In this framework, a schema is a similarity template for comparing subsets of strings with similarities at certain string positions. For example, the schema **01* matches the patterns 10011 and 00010. Consequently, the explicit processing of strings causes implicit processing of many schemata during each generation. To understand the growth and decay of the many schemata contained in a population, we introduce the following notations [13]. $\mathbf{A}(t)$ represents the population in generation t consisting of strings of length l each and $\overline{f}$ is the average fitness of the population; $o(H)$ is the order of a scheme H, i.e., the number of fixed positions in the template (number of 0s and 1s using a binary alphabet for the template); $\delta(H)$ is the defining length of H, i.e., the distance between the first and last specific string positions; $m(H,t)$ is the number of individuals of $\mathbf{A}(t)$ that match H and $f(H)$ is the average fitness of these strings. Assuming that the reproduction and crossover operations are mutually independent, the expected number of individuals matching H in the next generation under reproduction, crossover, and mutation is given by [13]

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\overline{f}} \left[1 - p_c \frac{\delta(H)}{l-1} - o(H)p_m\right] \tag{1}$$

where p_c and p_m are the crossover and mutation rates, respectively, and $p_m \ll 1$. The conclusion of (1) is that short (i.e., small $\delta(H)$), low-order (i.e., small $O(H)$), above-average (i.e., $f(H) > \overline{f}$) schemata receive exponentially increasing samples in subsequent generations.

GAs are able to overcome many of the problems mentioned earlier in the context of current 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.

The importance of GAs as function optimizers is well established and the topic has been the subject of several Ph.D. dissertations [1,3, 8–10 12, 27]. GAs are especially appropriate when the objective function is multimodal, high-dimensional, and/or contaminated by noise. (It is known that the GA performance is inferior to HC methods for unimodal, low-dimensional, and noiseless functions [1].) GAs allow the possibility of achieving the global maximum without exhaustive search. For example, during global optimization of gas pipeline operations [12], near optimal results were found after examining an infinitesimal fraction (10^{-6} to 10^{-7}) of the search space. Additionally, the performance of every GA is evaluated using an external evaluation procedure. By appropriately biasing the performance evaluation criteria, greater control over the genetic search process is possible than in other optimization techniques. On the other hand, optimizing functions, called GA-hard, tend to have remote, highly isolated optima and are difficult for any optimization technique (except exhaustive search).

B. Hybrid Search Techniques

GAs have been proven [13, 15] and shown to provide robust search performance across a broad spectrum of problems. However, hybrid techniques [1] have the potential for improved performance over single optimization techniques since these can exploit the strengths of the individual approaches in a cooperative manner. One such hybrid scheme which is the focus of this work combines a global search technique (GA) with a specialized local search technique (HC). HC methods are not suitable for optimization of multimodal objective functions, such as the segmentation quality surfaces, since they only lead to local extrema and their applicability depends on the contour shape of the objective functions. The hybrid scheme provides performance improvements over the GA alone by taking advantage of both the global search ability of the GA and the local convergence ability of the HC. In a sense, the GA first finds the hills and the HC climbs them.

III. MULTIOBJECTIVE OPTIMIZATION WITH GENETIC ALGORITHMS

A single-objective optimization problem is concerned with optimizing a single objective such as minimizing cost or maximizing profit. However, in many real-world applications, it is important that optimization techniques be capable of handling multiple noncommensurable objectives. In this section, we provide the basics of multiple objective optimization and discuss how a GA might be suitable for such problems.

A. Multiobjective Optimization

A multiple objective constrained optimization problem is of the form

$$\max[f_i(\mathbf{x}) = z_i], \qquad i = 1,\ldots,k, \quad \text{such that} \quad \mathbf{x} \in S$$

where $f_i(\mathbf{x})$s are the objective functions and z_is are the corresponding optimal criterion values and S is the feasible region. However, it is only in the trivial case, that there exists a single point in S which simultaneously maximizes all k objectives. A typical approach in multiobjective (or vector-valued) optimization is to consider the *utility* of the z_is. Thus, a point in S is optimal if it maximizes the decision maker's utility function. To be optimal, however, a point must be *efficient* or *Pareto* optimal [24].

The key concept of Pareto optimality is the "partially greater than" ($p >$) relation between two vectors of the same dimension. Given two vectors $\mathbf{a} = (a_1,\ldots,a_n)$ and $\mathbf{b} = (b_1,\ldots,b_n)$, $\mathbf{a}$ is said to be partially greater than $\mathbf{b}$ ($\mathbf{a}p > \mathbf{b}$) if each element of $\mathbf{a}$ is greater than or equal to the corresponding element of $\mathbf{b}$ and at least one element of $\mathbf{a}$ is strictly greater than the corresponding element of $\mathbf{b}$, i.e.,

$$(\mathbf{a}p > \mathbf{b}) \rightarrow (\forall\, i)(a_i \geq b_i) \quad (\exists\, i)(a_i > b_i).$$

Under these conditions, we say that $\mathbf{a}$ dominates $\mathbf{b}$ or $\mathbf{b}$ is inferior to $\mathbf{a}$. If a vector is not dominated by any other vector, it is said to be *nondominated* or noninferior.

In the multiobjective optimization context, if $(\mathbf{x}^0, \mathbf{z}^0)$ maximizes the utility function, then the point $\mathbf{x}^0$ is Pareto optimal and the criterion vector $\mathbf{z}^0$ is nondominated, i.e., not dominated by the criterion vector of some other point in S. In other words, it is not possible to move feasibly from $\mathbf{x}^0$ to increase an objective without decreasing at least one other objective, i.e., $\mathbf{z}^0$ is no longer nondominated. The set of all nondominated vectors is called Pareto-optimal set. The goal of a search for optima in a vector-valued space is, then, locating Pareto-optimal set. In practice, very often one is satisfied with a "near optimal" solution, one that is close enough to being optimal to be useful, to a multiobjective optimization problem.

B. Genetic Algorithm for Multiobjective Optimization

To show that a point is inefficient, one has to simply find another point in the feasible region whose criterion vector dominates that of the former. On the other hand, to show that a point is efficient or Pareto optimal requires an exhaustive test. It must be shown that none of the criterion vectors of other points in the feasible region dominates the criterion vector of the point in question. Recall that GAs work with a population of candidate solutions instead of a single solution. Thus, in the context of multiple objective optimization using GAs, the goal of the optimization is to identify an efficient schema. Since a single schema matches several individuals (i.e., points) in the genetic population belonging to the feasible region (refer to the discussions in Section II A), one associates a set of fitness (i.e., criterion) vectors, each of which corresponds to an individual, to a schema. Let $Z(H)$ denote the set of fitness vectors corresponding to the schema H.

DEFINITION. A schema H in S is *efficient* or *Pareto-optimal* if and only if (iff) there does not exist another schema $\overline{H}$ such that $Z(\overline{H})p > Z(H)$, $Z(\overline{H}) \neq Z(H)$.

Consequently, $Z(H)$ is the Pareto-optimal set if H is an efficient schema. According to (1), among all schemata of the same order and defining length, the most preferred schema is one whose average fitness is the highest. Now, the average fitness of a schema H is the average of the fitness values of the vector elements of $Z(H)$. Since the average fitness is maximum for a Pareto-optimal set in any given generation, the genetic evolution is biased towards an efficient schema. This property of a GA makes it ideal for multiobjective optimization problems.

The use of GAs in multiobjective optimization problems has been limited [20, 21]. In Schaffer [20], vector evaluated genetic algorithm (VEGA) creates equally sized subpopulations for selection along each of the criteria components in the fitness vector. The selection process is carried out independently for each criterion; however, reproduction and crossover are performed across subpopulation boundaries. Schaffer and Grefenstette [21] applied the VEGA to multi-class pattern discrimination problems which could not be solved by the single-objective GA. The problem with this latter mode of GA was that knowledge structures containing complementary knowledge were forced to compete by the GA using a scalar fitness function. In this work, the VEGA is used as a learning component to identify the knowledge structure that contains the most promising classification rule. The use of fitness vectors helps to overcome two major deficiencies of a scalar GA: the inability to identify promising rules in the early stages of the task when successes are rare, and the inability to distinguish the better rules in the latter stages when promising rules are abundant.

IV. ADAPTIVE SEGMENTATION ALGORITHM

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. 2. After acquiring an input image, the system analyzes the image characteristics and passes this information, in conjunction with the observed external variables, to the machine learning component (GA or GA-HC hybrid). Using this data, the machine 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

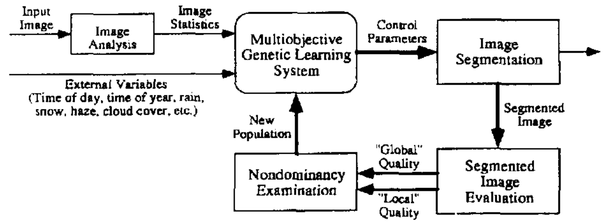


Fig. 2. Block diagram of adaptive image segmentation system for multiobjective optimization.

passed back to the learning subsystem. This process continues until a segmentation result of acceptable quality is produced. The details of each component in this procedure are described in the following subsections.

A. Image Characteristics

The inputs to the adaptive image segmentation system are color images of an arbitrary scene. These images must be analyzed so that a set of features can be extracted to aid in the parameter selection process performed by the machine learning component. A set of characteristics of an 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.

Corresponding to each input image, the system computes twelve statistics for each of the red, green, and blue components of the image. These statistics are based on the first-order image properties and histogram properties. They 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* [17, 23] 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. External variables 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. can also be used to characterize an input image. In our approach, two external variables, time of day and weather conditions, are used to characterize the image. These factors specify the conditions under which the image was acquired and provide useful information in representing the overall characteristics

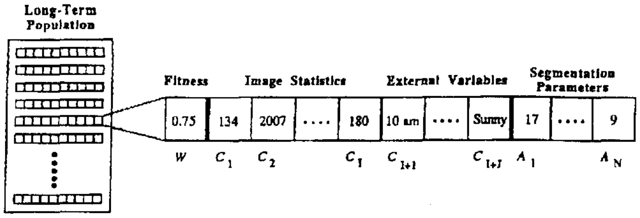


Fig. 3. Representation of knowledge structure used by genetic learning system. Image characteristics (image statistics and external variables), segmentation parameters, quality or *fitness* of parameter set stored in each knowledge structure.

of the input image. The variations in these image characteristics affect the quality of the image, which in turn necessitates changes in the control parameters of the segmentation algorithm. 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 variable values corresponding to two different images are computed symbolically when measuring image similarity. Addition of the two external variables to the list of image characteristics results in a 50 element list for the outdoor experiments.

B. Machine Learning Subsystem

Once the image statistics and external variables have been obtained, the machine learning subsystem uses this information to select an initial set of segmentation algorithm parameters. A database of knowledge structures is used to represent the image characteristics and the associated segmentation parameters. Fig. 3 shows an example of a knowledge structure. The knowledge structure 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. For the pure-GA approach, the fitness or the segmentation quality **W** is vector valued in local and global quality measures each of which ranges in value from 0.0 to 1.0. However, the fitness is scalar for the hybrid algorithm. In our experiments, the global quality measure is the weighted sum of edge-border coincidence and boundary consistency with each measure having equal weight. Similarly, the local quality measure is the weighted sum of pixel classification, object overlap and object contrast with each measure having the same weight. The evaluation measures are discussed in Section IV D. Only the fitness value(s) and the segmentation parameters of a knowledge structure are subject to genetic adaptation; the conditions remain fixed for the life of the knowledge structure. We have adopted two approaches for the adaptation process. The first one is based on pure genetics, while the second approach is a hybrid of genetics and HC. These are now described.

1) *Genetic Algorithm-Based Search*: Using the image statistics and external variables, the genetic learning system selects a small set (short-term population) of plausible parameter combinations from the current collection of parameter sets (long-term population). The long-term population represents the accumulated segmentation experience for all images that the system has processed, whereas the short-term population contains the set of segmentation parameters processed by the GA during the optimization for the current image. If the fitness values are not acceptable, the individuals are recombined and the process repeats. The generation cycle (segmentation, evaluation, recombination) continues until the stopping criterion is fulfilled. Each control parameter set in the final short-term population is then passed to the *Phoenix* algorithm. Fig. 2 schematically describes the adaptive image segmentation system based on multiobjective optimization of the segmentation quality measures. A description of the multiobjective optimization approach for image segmentation using a pure GA is given below.

1. Compute the image statistics.
2. Generate an initial population.
3. Segment the image using initial parameters.
4. Compute the global and local quality measures.
5. Examine nondominancy of each individual.
6. WHILE not ⟨stopping conditions⟩ DO
   - 6a. select subgroups of individuals using each dimension of the quality measures
   - 6b. generate new population using the crossover and mutation operators
   - 6c. segment the image using new parameters
   - 6d. compute the global and local quality measures
   - 6e. examine nondominancy of each individual

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

The process of selecting the short-term population in each generation is carried out for each dimension (i.e., global and local) of the segmentation quality vector and the number of individuals selected for each dimension is equal to the short-term population size divided by the vector size. The generation of a new population consists of the following steps: 1) select subgroups of individuals using each dimension of the quality measure in turn; 2) shuffle all the selected individuals; and 3) combine the individuals using crossover and mutation operators. This simple procedure ensures that any segmentation parameter which has above average performance on any quality measure is likely to survive and it gives appropriate selection preference to parameters that are above average on more than one quality measure.

After the image segmentation evaluation procedure has been applied, the nondominancy of each segmentation parameter is examined by comparing it with all other parameters in a population. It should be noted that this nondominancy or "dominate" test is strictly local. Pareto's concept of nondominancy implies comparison of a point to all other points in the search space, but our "dominate" test is limited to the current population. While a locally dominated point is also globally dominated, the converse is not necessarily true. A segmentation parameter which is nondominated in one generation may be dominated by a parameter which may emerge in a later generation. The "dominate" test is still useful because the set of nondominated parameters in each generation represents the current best guess of the Pareto-optimal set that will be improved in the future generations.

The stopping criteria for the multiobjective optimization system consist of two conditions. First, the process terminates if an utopian parameter set, i.e., the one for which both local and global quality measures are above a predefined threshold of acceptance, is located. The thresholds for acceptable segmentation is 90% of the best segmentation. This criterion is useful only when the best for each segmentation quality surface is known *a priori*. Second, the process terminates if both the average local quality and the average global quality of the populations decrease for three consecutive generations or fail to improve for five consecutive generations. If either of these conditions is met, the segmentation of the current image is stopped and the nondominated parameter sets are represented as the current best estimates of the Pareto-optimal set.

During the training phase, when there is no *a priori* knowledge available, the initial or *seed* population is created randomly. The objective of the training phase is to measure the optimizing parameters of the GA such as the convergence rate and the termination criterion. The knowledge base for the testing phase is created by collecting the final population corresponding to each training image to form the long-term population. During the testing phase, the seed population is created by selecting an initial set of segmentation parameters from the long-term population using the characteristics of the test images. Testing is carried out both sequentially and in parallel. In sequential testing experiments, the final population corresponding to each processed image is added to the long-term population to be used by the subsequent test images, unlike in parallel testing experiments where the results are not put back in the long-term population.

2) *Hybrid Search Combining Genetic Algorithm and Hill Climbing*: The hybrid search scheme is implemented in this research by switching control between the GA and the HC process according to simple transition rules. The block diagram of the adaptive image segmentation system using the hybrid optimization scheme is shown in Fig. 4. The image segmentation and evaluation components of Fig. 2 are grouped in one box in Fig. 4. According to these rules, the switch of control from the GA mode to the HC mode takes place when the GA finds a new maximum point and passes it to the hill climber as the starting point. Consequently, the hill climber passes the control over to the GA when it reaches a local maximum, a point that is better than all of its adjacent points. The local maximum point replaces the maximum point in the current population that has been the starting point for the hill climber, and the GA proceeds with the updated population. A description of hybrid algorithm is now given.

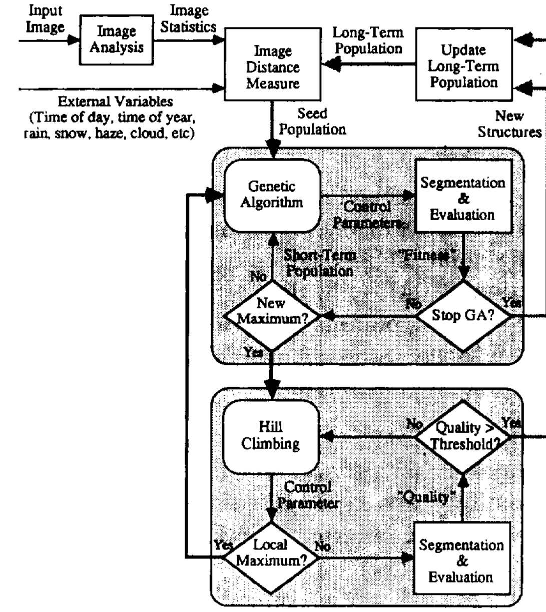


Fig. 4. Block diagram of adaptive image segmentation system using hybrid search scheme.

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

    IF ⟨new maximum found⟩ /* Hill climbing */

5HCa. generate all points (i.e., parameters) adjacent to the current point

5HCb. segment the image using adjacent parameters

5HCc. compute the segmentation quality measures

5HCd. climb to new maximum point if it exists

    ELSE /* Genetic algorithm */

5GAa. select individuals using the reproduction operator

5GAb. generate new population using the crossover and mutation operators

Mt2 □ job no. 1839 □ ieee □ aerospace and electronic systems □ 1835d03 [8] □ (XXX) □ 08-09-95 □ 9:31 am

5GAc. segment the image using new parameters
5GAd. compute the segmentation quality measures
END
6. Update the knowledge base using the new knowledge structures.

The search through a space of parameter values using HC consists of the following steps [1]. 1) Select a starting point. 2) Take a step in each of the fixed set of directions. 3) Move to the best alternative found. 4) Repeat until a point is reached that is higher than all of its adjacent points. An algorithmic description of the HC process is as follows.

1a. Select a point x_c at random.
1b. Evaluate the criterion function, i.e., obtain $V(x_c)$.
2a. Identify points $x_1,\ldots,x_n$ adjacent to x_c.
2b. Evaluate the criterion function, i.e., obtain $V(x_1),\ldots,V(x_n)$.
3. Let $V(x_m)$ be the maximum of $V(x_i)$ for $i = 1,\ldots,n$.
3a. If $V(x_m) > V(x_c)$ then
    set $x_c = x_m$, $V(x_c) = V(x_m)$
    goto Step 2.
3b. Otherwise, stop.

In the above, a set of points that are "adjacent" to a certain point can be defined in two ways. First, it can denote the set of points that are a Euclidean distance apart from the given point. Thus, the adjacent points are located in the neighborhood of the given point. Second, adjacent points can denote the set of points that are unit Hamming distance apart from the given point. Each point in this set differs by only one bit value from the given point in binary representation of points. It defines the set of points with varying step size from the given point. The set of Hamming adjacent points was used in this research. Hamming adjacent points have an advantage over Euclidean adjacent points in our implementation because all the segmentation parameter values are represented as binary strings when using the GA. The set of Hamming adjacent points also represents the set of points which can be generated by a genetic mutation operator from the given point.

A conventional HC approach, as described above, finds the largest $V(x_m)$ from $V(x_i)$, $i = 1,\ldots,n$, and the search moves to its corresponding point, x_m. For a space of n adjacent points, it requires n function evaluations to make each move. To reduce the cost of evaluating all the adjacent points before making each move, the hybrid approach is designed to try alternatives only until an uphill move is found. The first uphill move is undertaken without checking whether there are other (higher) possible moves. After the HC process has examined all the adjacent points by flipping each bit in the binary representation of the current point, in turn, without finding an uphill move, the current point is taken as a local maximum and the process passes the control to the genetic algorithm. The algorithmic description of the HC process used in the hybrid search scheme is as follows.

1. Select a starting point x_c with fitness value $V(x_c)$ from the genetic population.
2. Set $i = 0$.
3. Set $j = i$.
4a. Generate an adjacent point x_a by flipping the ith bit in x_c.
4b. Obtain $V(x_a)$. Set $i = (i+1) \bmod n$.
5. If $V(x_a) > V(x_c)$ then
    set $x_c = x_a$
    goto Step 3.
   Else if $i < j$ then
    goto Step 4
   Otherwise, pass the control to the GA.

The GA employed in the hybrid approach is the same as the one used in the pure genetics approach. The termination criteria for the hybrid search process consist of three conditions. First, the process terminates when either the GA or the hill climber finds a parameter set with a segmentation quality equal to or higher than a predefined threshold. The threshold (90%) is chosen to be the same as in the pure genetics case. Second, the GA terminates if the average fitness value of the short-term population continuously decreased for three consecutive generations or failed to improve for five consecutive generations. Third, the GA terminates after 50 generations. This condition is included only to ensure the termination of the algorithm. If any one of these three conditions is met, the processing of the current image is terminated and the long-term population is updated using the high quality individuals in the short-term population.

C. Image Segmentation

The image segmentation component is the *Phoenix* algorithm [17, 19, 23] which has been extensively tested on color imagery and has been assimilated into the DARPA/SRI Image Understanding Testbed [17]. 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. 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 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 would permit.

In the histogram analysis phase, *Phoenix* computes a histogram for each feature plane, analyzes it and selects thresholds or histogram cutpoints (a set of thresholds is called an interval set) which are likely to isolate significant homogeneous regions in the image. Initially, the histogram is smoothed with an unweighted window average, where the width of the window is specified by the *hsmooth* threshold. It is then broken into intervals such that each begins with a valley, contains a peak, and ends on the next valley. (A valley is considered as the right and left "shoulders" of its two surrounding intervals.) An interval is retained if the ratio of the peak height to the height of its higher shoulder, expressed in percentage, is greater than or equal to the *maxmin* threshold. There are additional tests to decide whether an interval should be retained or not; however, these are not discussed due to space limitations. When an interval is eliminated, it is merged with the neighbor sharing the higher of its two shoulders. The process of merging intervals with low peak-to-shoulder ratio is continued until the number of intervals reach a prespecified limit. A score is also computed for each interval in the set. The interval set score is the maximum of all the interval scores.

The spatial analysis phase, following the histogram analysis, selects the interval sets with highest scores (one set per feature plane), thresholds the corresponding feature planes and extracts connected components for spatial evaluation. The histogram cutpoints are now applied to the feature plane as intensity thresholds and connected components are extracted. Patches smaller than noise pixels are considered to be "noise" regions. After each feature has been evaluated, the one producing the least total noise area is accepted as the segmentation feature, provided that the total noise area is less than certain percentage of the total region area. If no suitable feature is found, the original region is declared terminal. Otherwise the valid patches, merged with the noise patches, are converted to new regions and added to the segmentation record. In either case, a new segmentation pass is scheduled next.

Phoenix contains seventeen different control parameters [17], fourteen of which are used to control the thresholds and termination conditions of the algorithm. There are 10^{33} conceivable parameter combinations using these fourteen values. Of the fourteen values, we have selected two of the most critical parameters that affect the overall results of the segmentation process: *maxmin* and *hsmooth*. From an analysis of the *Phoenix* algorithm, we find that incorrect values in the two main 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. By measuring segmentation performance using appropriate quality criteria, the genetic process attempts to identify a parameter set that yields results between these two extremes.

D. Segmentation Evaluation

After the image segmentation process has been completed by the *Phoenix* algorithm, the overall quality of the segmented image must be measured. There are a large number of segmentation quality measures that have been suggested in the literature [4], 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 described below to determine the overall fitness for a particular parameter set. 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 are as follows.

1) *Edge-Border Coincidence*: Measures the overlap of the region borders in the segmented image with the edges found in the original image using an edge operator (Sobel operator). Let E be the set of unthinned edge pixels and let S be the set of boundary pixels. Thus,

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

edge-border coincidence

$$= \frac{n(E \cap S)}{n(E)}$$

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

Here, $n(A)$ denotes 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 (unthinned) information. For each pixel in the boundary pixel set S, a neighboring pixel in the edge pixel set 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

$$R_i = \frac{d_{\max} - d_i}{d_{\max}}, \qquad \text{where} \quad d_{\max} = 10,$$

and d_i = the distance to the nearest edge pixel. Thus, if the boundary and edge pixels had overlapped, $R_i = (10-0)/10 = 1$. Pixels that do not directly overlap contribute a reward value that is inverserly related to their distance from each other. As matching pairs of pixels are identified, they are removed from the region boundary and edge pixel sets (i.e., from 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., nonoverlapping and nonneighboring) pixels correspond to the difference between the two images. These pixels are used to compute a penalty as follows

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

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 A be the set of object pixels in the groundtruth image and B be the set of object pixels in the segmented image. Formally, we have

$$\begin{aligned} A &= \{p_1, p_2, \ldots, p_A\} \\ &= \{(x_{p1}, y_{p1}), (x_{p2}, y_{p2}), \ldots, (x_{pA}, y_{pA})\} \\ B &= \{q_1, q_2, \ldots, q_B\} \\ &= \{(x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \ldots, (x_{qB}, y_{qB})\}. \end{aligned}$$

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

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

where $A \cap B = \{(x_k, y_k),\ k = 1, \ldots, m \text{ where } (x_k, y_k) \in A \text{ and } B\}$. Using the value of N, pixel classification can then be computed as

$$\text{pixel classification} = N, \qquad \text{if} \quad N \geq 0,$$

and zero otherwise.

4) *Object Overlap*: Measures the area of intersection between the object region in the groundtruth image and the segmented image. Once again, let A denote the set of object pixels in the groundtruth image and B denote the set of object pixels in the segmented image. Then, object overlap can be computed as

$$\text{object overlap} = \frac{n(A \cap B)}{n(A)}$$

where $A \cap B = \{(x_k, y_k),\ k = 1, \ldots, m \text{ where } (x_k, y_k) \in A \text{ and } B\}$.

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 A and B be the sets of object pixels in the groundtruth and the segmented images, respectively. 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 (A and B) 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 A and B object regions.

We compute the average intensity, $I_R = (\sum_{j=1}^{R_{\max}} I(j))/R_{\max}$, for each of the four regions A, B, X, and Y, where $I(j)$ is the intensity of the jth pixel in some region R, and $R_{\max}$ is the total number of pixels in region R. 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

$$C_{GT} = \left|\frac{I_A - I_X}{I_A}\right|, \qquad C_{SI} = \left|\frac{I_B - I_Y}{I_B}\right|.$$

The object contrast quality measure is then computed as

$$\text{object contrast} = \begin{cases} \dfrac{C_{SI}}{C_{GT}}, & \text{if} \quad C_{GT} \geq C_{SI} \quad \text{or} \\[2ex] \dfrac{C_{GT}}{C_{SI}}, & \text{if} \quad C_{GT} < C_{SI}. \end{cases}$$

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, i.e., edge-border coincidence and boundary consistency, 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.

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

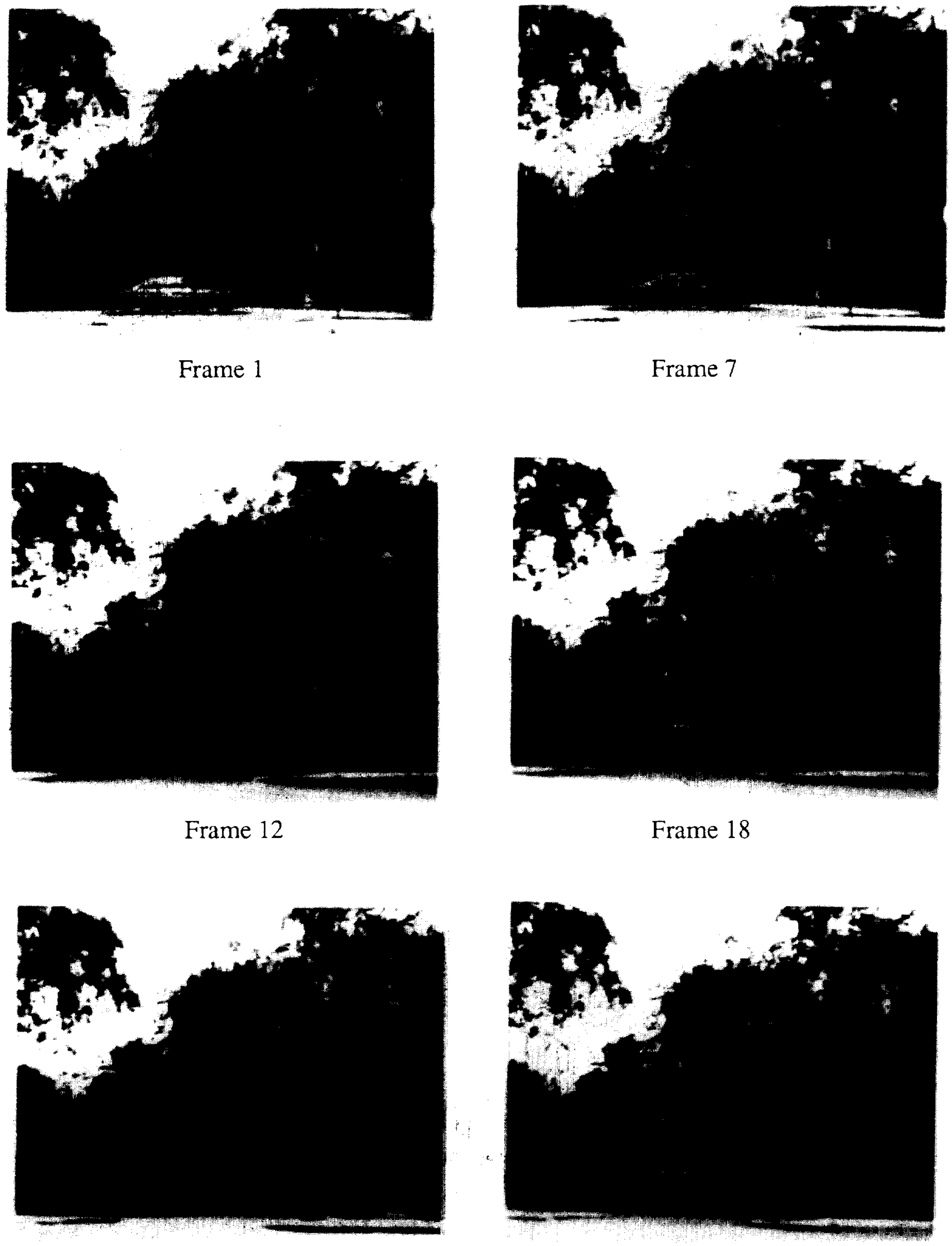


Fig. 5. (a) Sample frames of outdoor image sequence.

global and local quality measures together can be used to obtain a *scalar-valued* or a *vector-valued* segmentation quality measure that maximizes overall performance of the system. The maximization of the vector-valued segmentation quality measure is in effect a multiobjective optimization problem where the global and the local measures represent the "noncommensurable" criterion functions.

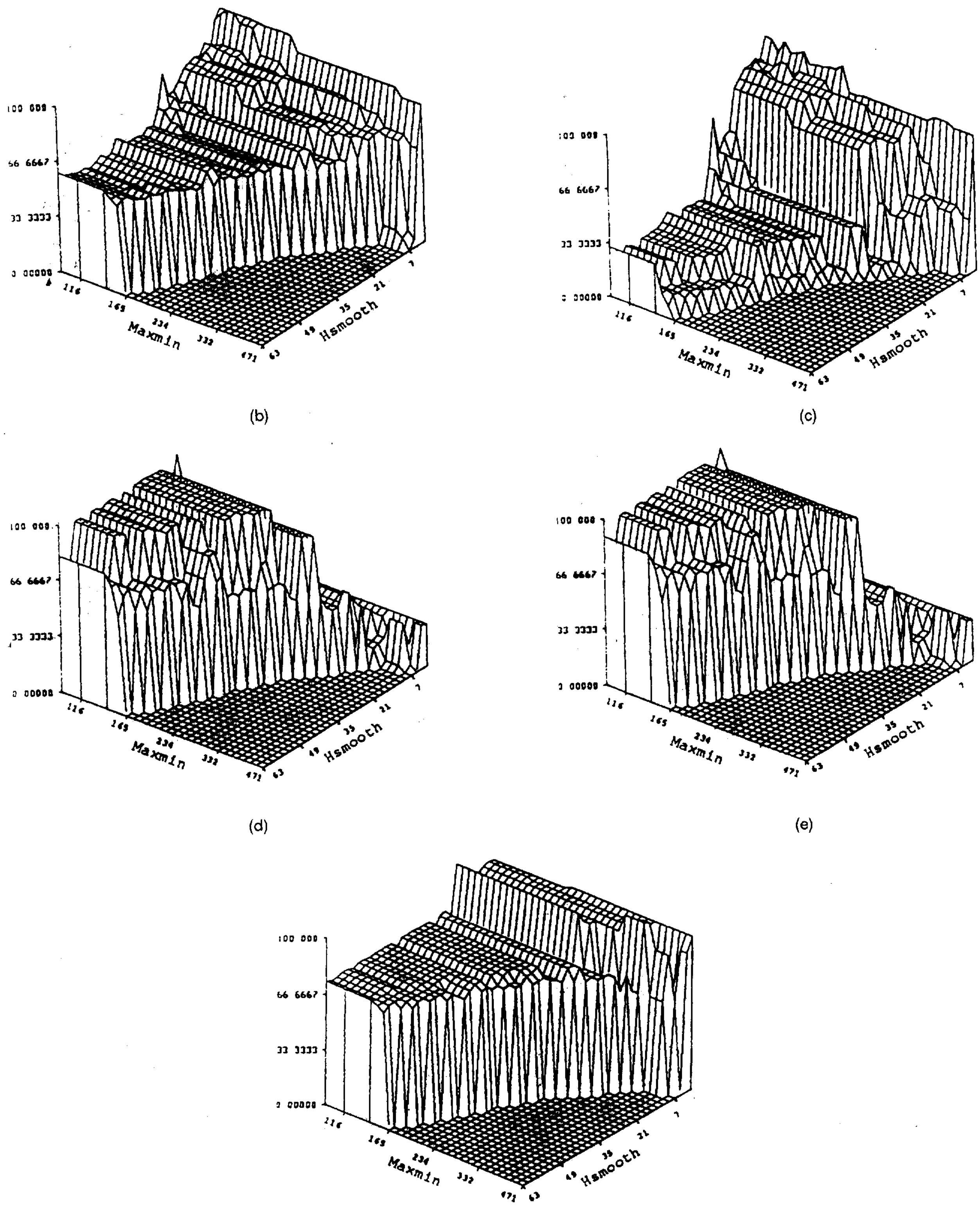
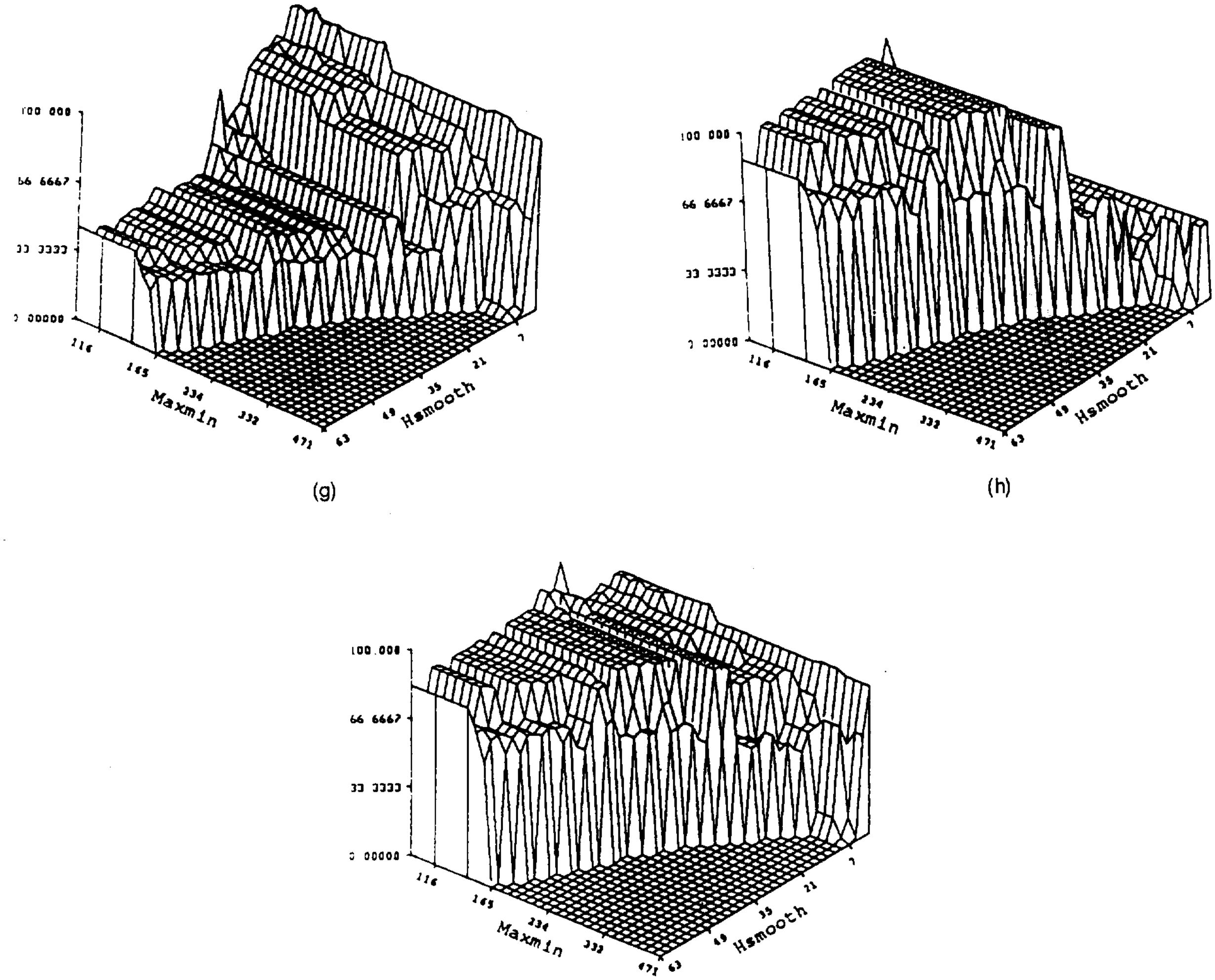


Fig. 5. (b)–(f) Segmentation quality measures for frame 1. (b) Edge-border coincidence. (c) Boundary consistency. (d) Pixel classification. (e) Object overlap. (f) Object contrast.

V. EXPERIMENTAL RESULTS

A database of 20 outdoor images of a static scene, collected over a 5 h period, is used by the system. The original images are digitized at 480×480 pixels in size and are then subsampled (average of 4 by 4 pixel neighborhood) to produce 120 by 120 pixel images for the segmentation experiments. Fig. 5(a)



(g) (h) (i)

Fig. 5. (g)–(i) Segmentation quality measures for frame 1. (g) Global. (h) Local. (i) Combined.

shows frames 1, 7, 12, 18, 19, 20 of 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 is obtained by manual segmentation of frame 1 of the image sequence. The segmentation quality surfaces, both global and local, for each frame are exhaustively defined for preselected ranges of *maxmin* and *hsmooth* parameters of the *Phoenix* algorithm. Default values are used for the remaining parameters. Figs. 5(b) to 5(f) show the segmentation quality surfaces for edge-border coincidence, boundary consistency, pixel classification, object overlap, and object contrast for the frame 1 shown in Fig. 5(a). Figs. 5(g), 5(h) and 5(i) show global, local, and combined quality surfaces for frame 1. The combined quality surface for frame 1 is a combination of equally weighted five quality measures shown in Figs. 5(b) to 5(f).

The outdoor image sequence is separated into two halves, 10 images for training and 10 images for testing. To ensure that the training and testing imagery are representative of the entire outdoor image sequence, the odd-numbered images (1,3,...,19) are selected as the training data, while the even-numbered images (2,4,...,20) are used for testing. Note that these images exhibit diversity in the environmental conditions which varied from bright sun to overcast sky. There are significant variations from image to image due to the changing position and intensity of the sun. This movement creates varying object highlights, moving shadows, and many subtle contrast changes. Colors of most objects in the image are subdued. The auto-iris mechanism in the color video camera (JVC GXF700U) was functioning. 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. This type of image data simulates a photointerpretation scenario in which the camera position is fixed and the image undergoes significant change over time.

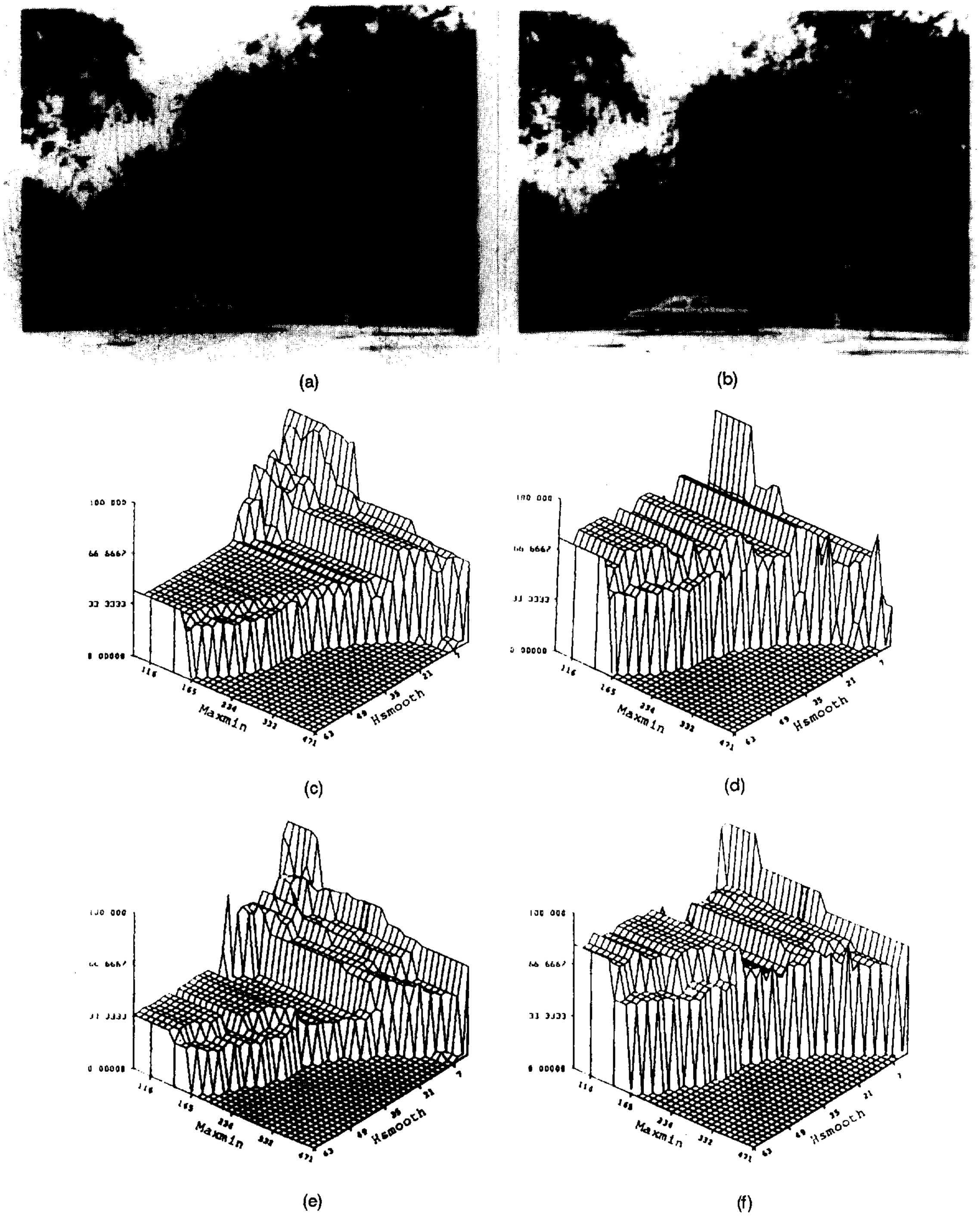


Fig. 6. (a) and (b) Color images for outdoor experiments (frames 3 and 4). Frames have been selected as representative frames of outdoor sequence for demonstrating results. (c)–(f) Global and local segmentation quality surfaces for frames 3 and 4. (c) Frame 3 global quality. (d) Frame 3 local quality. (e) Frame 4 global quality. (f) Frame 4 local quality.

We have selected frames 3 and 4 shown in Figs. 6(a) and 6(b) as the representative frames of the outdoor imagery for demonstrating results in this work. The global and local quality surfaces for these frames are shown in Figs. 5(c) to 5(f). The genetic component uses a long-term population size of 100 individuals, a short-term population size of 10, a crossover rate of 0.8, and a mutation rate of 0.01. The stopping criteria is 90% of the global and local maxima of the global and local quality surfaces of each image in the database.

A. Genetics-Based Results—Multiobjective Optimization

Each training image is processed 100 times, each with a different (randomly selected) seed population. Frame 3 is selected to describe the

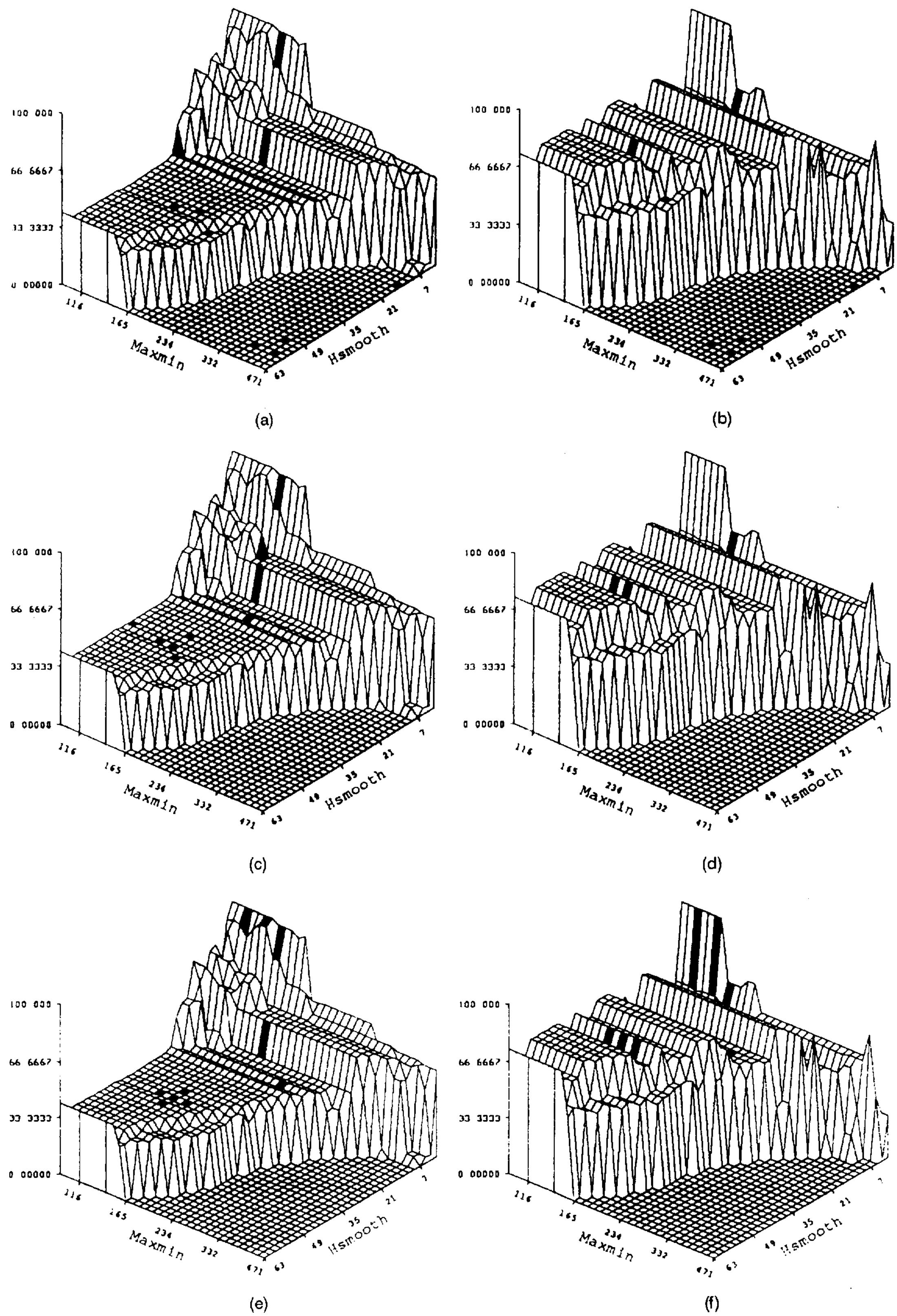


Fig. 7. Search point locations visited at each generation for frame 3. (a) Generation 1 global quality. (b) Generation 1 local quality. (c) Generation 2 global quality. (d) Generation 2 local quality. (e) Generation 3 global quality. (f) Generation 3 local quality.

representative experimental results of training during the multiobjective optimization. Fig. 7 illustrates the genetic search process at different generations using frame 3. The individuals of the short-term population are indicated on the quality plane in Fig. 8. The upper right corner of the plane represents the utopian

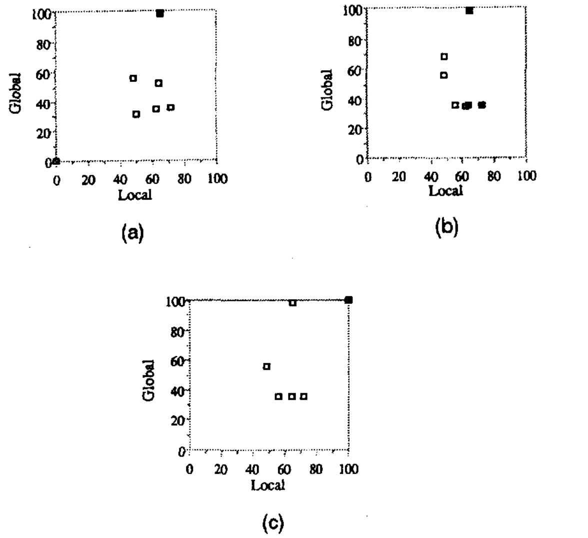


Fig. 8. Global and local segmentation quality of each individual at each generation for frame 3 of outdoor image database. Dark squares represent locally nondominated points at each generation. (a) Generation 1. (b) Generation 2. (c) Generation 3.

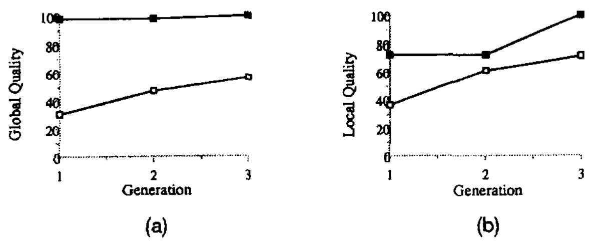


Fig. 9. Maximum and average fitness values of global and local quality measures at each generation for representative frame. Maximum fitness is indicated by top (dark-squared) line and average fitness is indicated by bottom line in these graphs. (a) Global performance. (b) Local performance.

point which has the maximum fitness values in both dimensions (i.e., global and local) of the segmentation quality measure. The squares in these plots represent the locally nondominated points at each generation. The plotted planes have less than 10 points (the current population size), because some individuals in the population have the same fitness values and are plotted at the same location. As an example, the four points located at the bottom plateau of the surfaces (i.e., global and local segmentation quality of 0.0) in Figs. 7(a)–(b) are plotted at the lower-hand corner of the graph in Fig. 8(a). Fig. 8(c) displays the utopian point at the upper right corner, which caused the termination of the genetic search process after third generation. In this figure, segmentation performance over 90% is denoted as 100%. Fig. 9 shows the maximum and average fitness values of the global and local quality measures corresponding to the generations referred to in Fig. 7.

Fig. 10 displays the segmentation results for frame 3. These results were obtained from the individuals in the short-term population with maximum global fitness

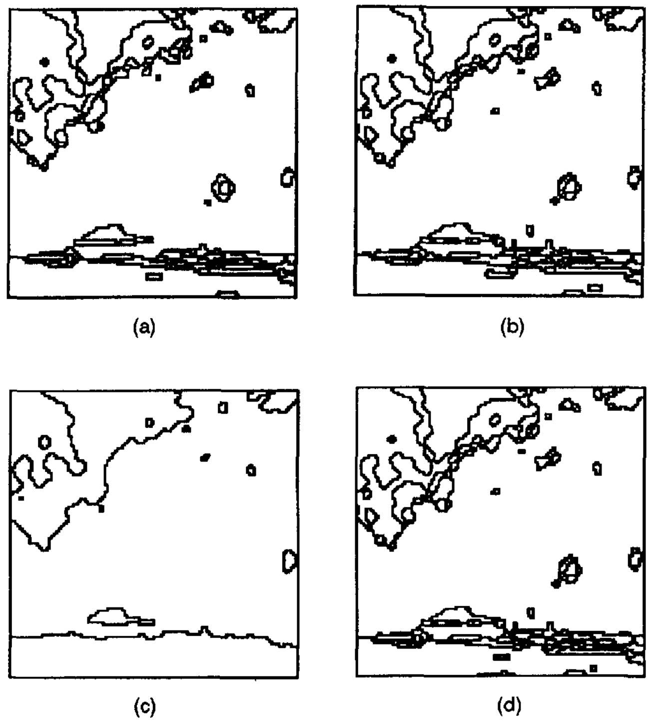


Fig. 10. Segmented images corresponding to frame 3. (a) Initial global results. (b) Final global results. (c) Initial local results. (d) Final local results.

(i.e., the best global segmentation quality) or maximum local fitness (i.e., the best local segmentation quality). An increase in overall segmentation quality between the initial and final results can be seen in these figures. The global segmentation result, which optimized the segmentation quality measure of the whole image, shows more precise boundary representations for all objects including the car. The local segmentation result, which optimized the segmentation quality measure of the object regions of interest (i.e., the car in this case), indicates that the portion of the car extracted from the image becomes larger in the final results. Notice that the bottom of the car is extracted as a separate region from the background in the final result, although this region is still not combined with the top portion of the car to form a single region. The performance of the multiobjective training experiments is summarized in Fig. 11. According to these plots, the maximum number of generations has been 10, the minimum number is 2, and the average number is 5.6. The small number of generations corresponding to frame 1 and frame 3 are caused by utopian points which perform well in both the global and the local measures.

During testing, the seed population is selected from the long-term population obtained at the end of the training experiments. Since the fitness values of the testing seed population are usually high, the GA converged to the Pareto-optimal set much faster during the testing experiments than

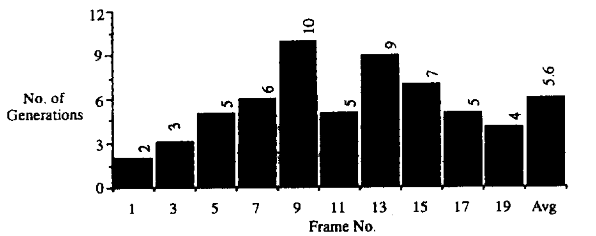


Fig. 11. Summary of performance for multiobjective optimization training experiments. Starting with set of random seed points on quality surfaces, adaptive image segmentation system optimized global and local quality measures of each image in number of generations indicated in graph.

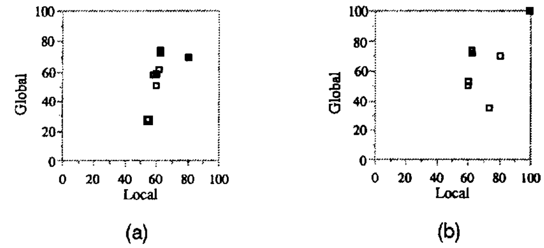


Fig. 13. Global and local segmentation quality of each individual at each generation for frame 4 during testing experiments. (a) Generation 1. (b) Generation 2.

in the training experiments. Fig. 12 shows the genetic search process at different generations for the test image of frame 4. The global and local quality measures of the short-term population in these generations are shown in Fig. 13 and the corresponding maximum and average values are indicated in Fig. 14. Frame 4 segmentation results using the parameters corresponding to the initial and final maximum values of the global and local measures are displayed in Fig. 15. The improved quality of the initial segmentation results during testing can be visually compared with the initial results acquired during training (Fig. 10). As with the training results, the segmentation quality in each of these images is the best possible result available using the *Phoenix*

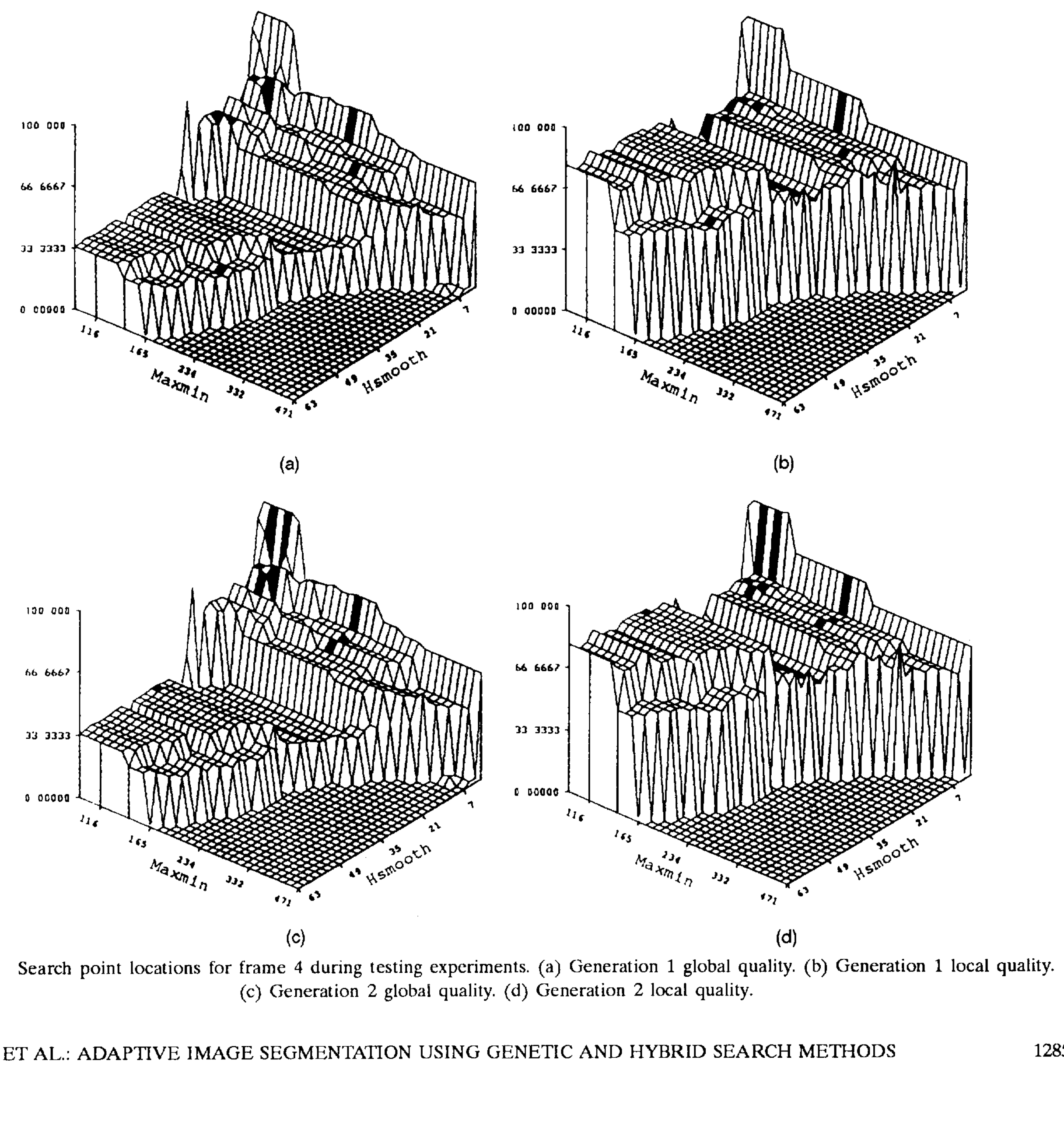


Fig. 12. Search point locations for frame 4 during testing experiments. (a) Generation 1 global quality. (b) Generation 1 local quality. (c) Generation 2 global quality. (d) Generation 2 local quality.

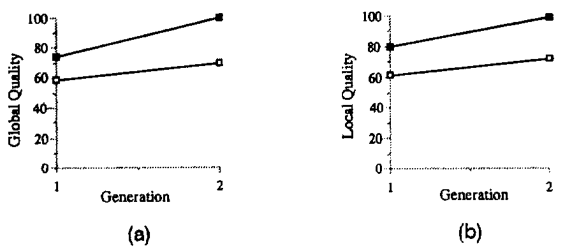


Fig. 14. Maximum and average fitness values of global and local quality measures at each generation for frame 4. Maximum fitness is indicated by top (dark-squared) line and average fitness is indicated by bottom line in these graphs. (a) Global performance. (b) Local performance.

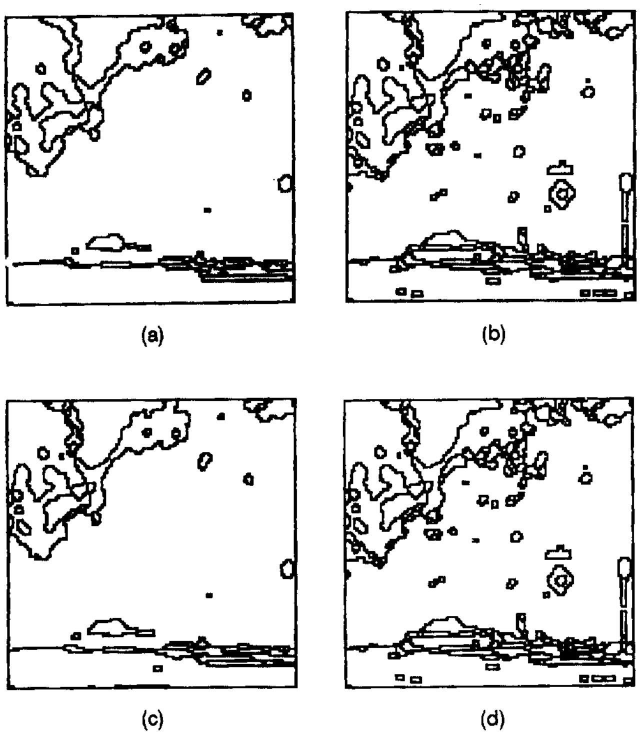


Fig. 15. Segmented images corresponding to frame 4. (a) Initial global results. (b) Final global results. (c) Initial local results. (d) Final local results.

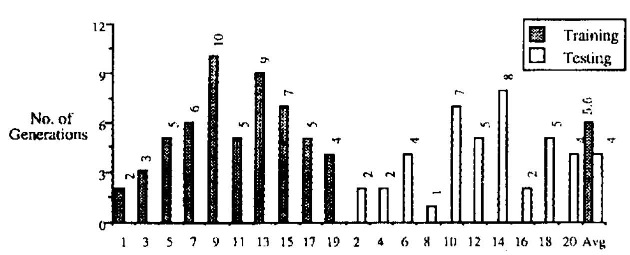


Fig. 16. Performance comparison of training and testing experiments for multiobjective optimization. By using experience stored in long-term population obtained from training experiments, adaptive segmentation system reduced the effort needed to optimize segmentation quality measures during testing phase.

algorithm. Fig. 16 compares the performance of the adaptive system during the training and testing phases. The average number of generations shows that the training results reduce the search efforts during testing. Overall, the simultaneous optimization of the global and local segmentation quality measures results in promising performance for the adaptive system. Elsewhere, we have compared the performances of GA and random search and have shown the effectiveness of the crossover and mutation operations which demonstrate that the genetics-based search is useful [6].

B. Hybrid Search-Based Results and Comparison with Pure Genetics

The same outdoor imagery database is used for the hybrid algorithm. Also, the training and testing sequences are kept unchanged. Recall that the fitness is now a scalar, combining the global and local segmentation quality measures, for each individual of the genetic population. In the training experiments, the hybrid search process was invoked with randomly selected locations on the combined surface for each training image and the convergence rate of the process was measured. Also, as in the baseline experiments (pure genetics), each training image was processed 100 times, each with a different collection of random starting points. The stopping criteria are discussed in Section IV B2.

To describe the complete training results of the hybrid scheme, frame 3 of the outdoor image database (Fig. 5(a)) is again selected as the representative frame. The progression of the hybrid search process at each generation corresponding to frame 3 is illustrated in Fig. 17. Each individual in the short-term population is plotted on the quality surfaces for the six generations that were necessary to optimize the segmentation quality. Generations 2 and 6 were processed by the hill climber after the GA produced the new maximum points in the previous generations. Notice that the movement of the maximum point (the rear corner point in Fig. 17(f)) between Generations 5 and 6 is due to the HC search process. Fig. 18 indicates the maximum and average fitness values of the short-term population during each generation. It is seen that the maximum fitness values increase continuously because the best individual in the population is always retained from one generation to the next. Average fitness values, on the other hand, fluctuate as the individuals visit different regions of the surface in search of highly fit areas. However, they increase gradually as the hybrid search process progresses.

To provide a visual indication of the performance improvements achieved by the adaptive segmentation system using the hybrid search scheme, the segmented image results are shown for frame 3 in Fig. 19. These results are obtained using the individual from the

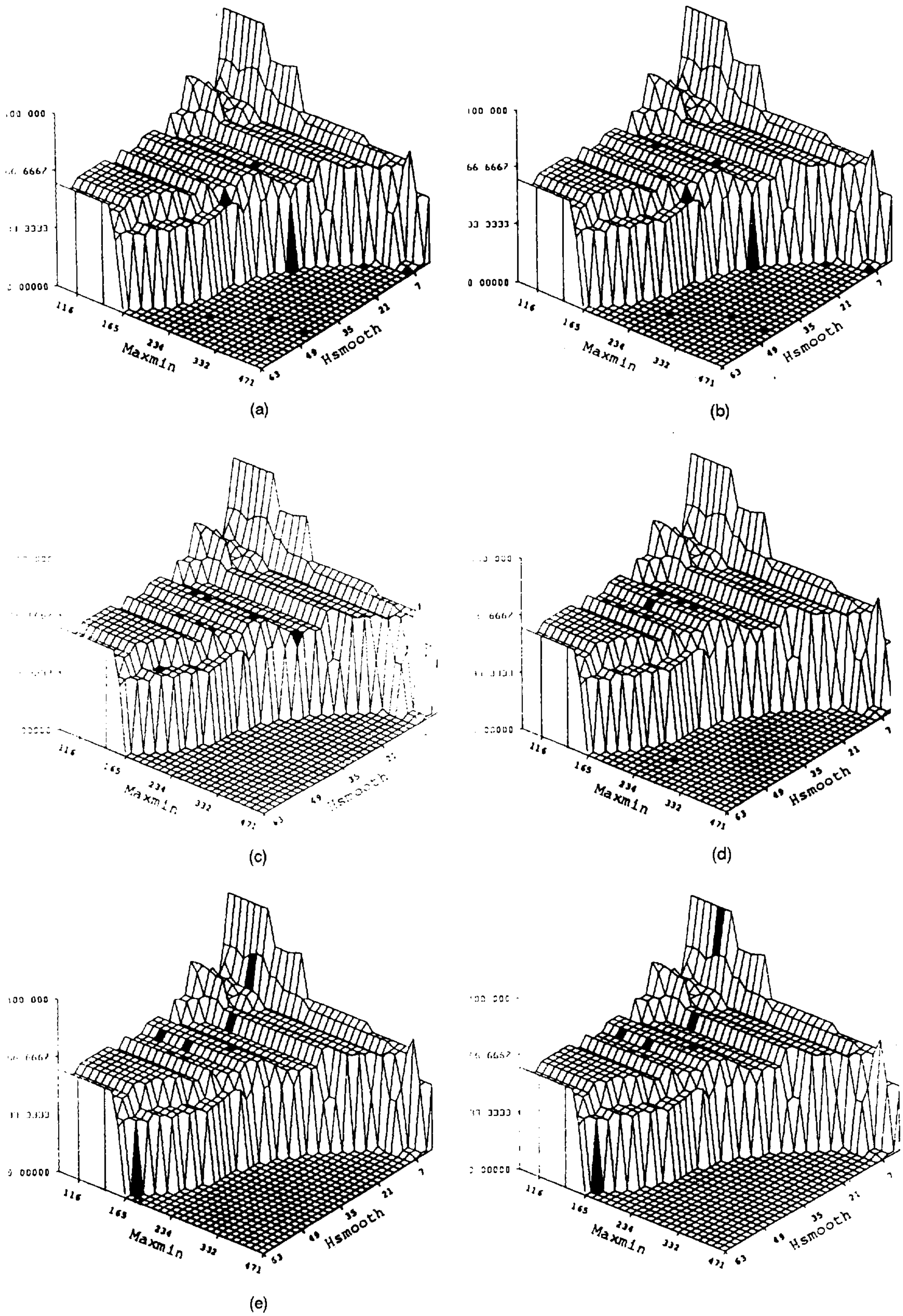


Fig. 17. Search point locations visited at each generation for frame 3 in hybrid experiments. (a) Generation 1. (b) Generation 2. (c) Generation 3. (d) Generation 4. (e) Generation 5. (f) Generation 6.

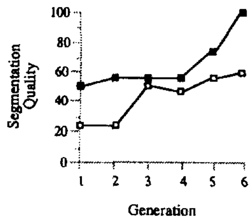


Fig. 18. Maximum and average fitness values of combined quality measures at each generation for frame 3 in hybrid experiments. Maximum fitness is indicated by top (dark-squared) line and average fitness is indicated by bottom line in these graphs.

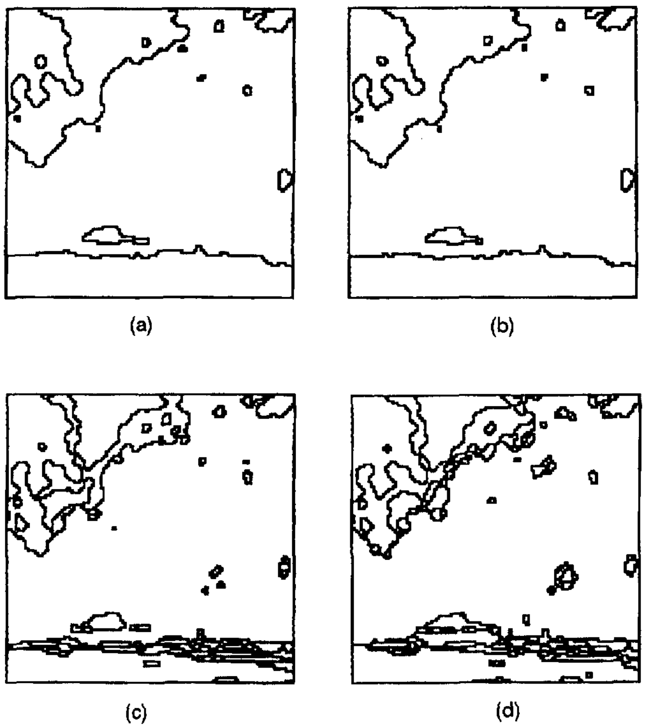


Fig. 19. Segmented images corresponding to frame 3 in hybrid experiments. (a) Generation 1. (b) Generation 2. (c) Generation 5. (d) Generation 6.

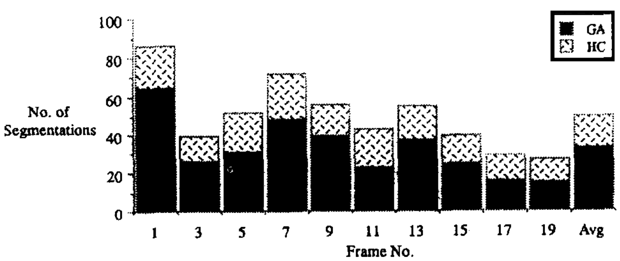


Fig. 20. Summary of performance during training experiments of hybrid scheme. Total number of segmentations required by GA and HC to optimize segmentation quality of each image are indicated in graph.

short-term population that has the maximum fitness (i.e., correspond to the best segmentation quality). Each of these segmented images shows a tendency to obtain more precise boundary representations for each of the background objects as well as the border of the car. The performance of the system on the

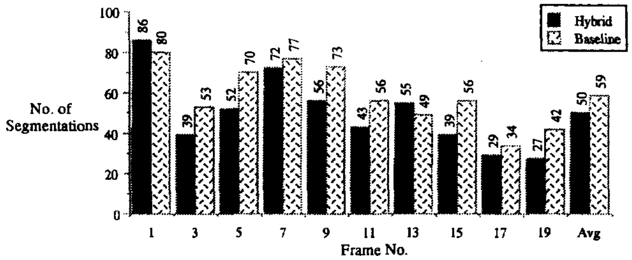


Fig. 21. Performance comparison of hybrid scheme and pure genetics training experiments. By combining GA with HC, the hybrid scheme reduces average computation effort needed to optimize segmentation quality of outdoor imagery.

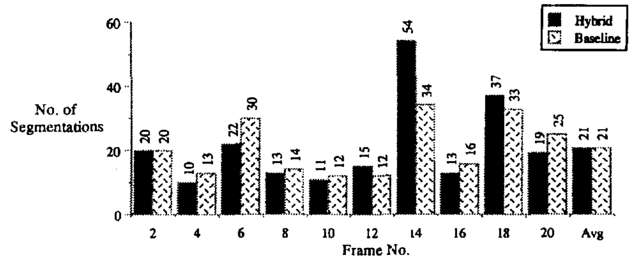


Fig. 22. Performance comparison of hybrid scheme and pure genetics training experiments. Hybrid scheme shows no clear-cut performance improvement over pure genetic approach because training results provide highly fit seed points for testing experiments in both schemes.

training images is illustrated in Fig. 20. It shows the total number of segmentations needed to optimize the segmentation quality of each image as sum of those required by the GA and the HC, individually. The maximum number of segmentations for these images was 86, the minimum number was 27, and the average numbers of segmentations required by the GA, the HC and the hybrid processes were 33, 17, and 50, respectively. Fig. 21 compares the performance of the hybrid scheme experiments with that of the baseline experiments. To ensure the fairness of the performance comparison, the computational efforts were measured by the number of segmentations (i.e., the number of points visited on a segmentation quality surface) required by the search processes. The hybrid scheme results surpassed the baseline results in 8 (out of 10) training images when the number of segmentations required to optimize the segmentation quality was reduced. On the average, an improvement of 15.3% in performance was observed with the training images.

The testing phase of the hybrid scheme experiments was also conducted similar to the baseline experiments. As in the baseline experiments, it was designed to measure the reduction in effort obtained by initializing the hybrid search process with nonrandom starting points. The long-term population (100 individuals) for each testing image was created by collecting the final short-term population of all the training images. The testing seed population (10 individuals) for each image was then selected from its own long-term population. Fig. 22 compares the performance of the hybrid scheme experiments with that of the baseline experiments during testing. The hybrid scheme results are no better than the baseline results, because the training results supplied the testing seed points located in highly fit regions of the search space which could hardly be optimized by the HC process. In summary, the hybrid search process performed well for the training experiments which proceed with random starting points. However, it could not improve performance over the pure GA for the testing experiments which proceeded with the highly fit starting points. In general, the hybrid scheme performs better than the pure GA for the frames which require less computational effort to optimize the segmentation quality, i.e., for the frames which have simpler segmentation quality surfaces.

VI. CONCLUSIONS

This paper has been concerned with an adaptive approach to the important low-level vision problem of image segmentation. It argued that adaptation is an important characteristic of an intelligent system and is essential for future computer vision systems which must operate in dynamic outdoor environments. The adaptive image segmentation system described incorporates a feedback loop consisting of a machine learning subsystem, an image segmentation algorithm, and an evaluation component which determines segmentation quality. The machine learning component is based on genetic adaptation and uses (separately) a pure genetic algorithm and a hybrid of GA and HC. When the learning subsystem is based on a pure GA, the corresponding evaluation component is based on a vector of evaluation criteria. For the hybrid case, the system employed a scalar measure which is a weighted combination of the different criteria. We have presented the detailed experimental results for pure genetic and hybrid search methods using a representative database of outdoor TV imagery. The multiobjective optimization demonstrates the ability of the adaptive image segmentation system to provide high quality segmentation results in a minimal number of generations. The results of the hybrid method show the performance improvement over the GA alone. Elsewhere we have shown the ability of the adaptive image segmentation (with pure GA and scalar evaluation measure) that the adaptive segmentation system provides high quality ($> 95\%$) segmentation results in a minimum number of segmentation cycles. It consistently performs better than the default parameters or the traditional techniques commonly used in image processing and computer vision. Further, we have shown that learning from experience can be used to improve the performance of the system with time [7].

The adaptive image segmentation system can make use of any segmentation algorithm that can be controlled through parameter changes. No extensive knowledge pertaining to the selected algorithm is required. In addition, one can choose to adapt the entire parameter set or just a few of the critical parameters, depending on time constraints and the desired quality of the final segmentation results. Although only color images have been used in this work, the adaptive technique itself is applicable to any type of imagery whose characteristics can be properly represented. This set includes infrared, laser radar, millimeter wave, sonar, and gray-scale imagery. The adaptive image segmentation system can utilize local, global, or combined segmentation quality measures to achieve the appropriate segmentation results. If we need a highly accurate segmentation of specific object regions only, the local quality measures can be used to produce this result. Conversely, if we require a good segmentation of the entire image, the global quality measures allow us to obtain this results. The combined quality measures provide intermediate segmentation results between these two extremes. However, it may be noted that to use any of the segmentation results presented in this work for higher level processing such as object recognition some form of grouping of the labeled region pixels will be necessary.

The adaptive image segmentation system is only as robust as the segmentation algorithm that is employed. It cannot cause an algorithm to modify the manner in which it performs the segmentation task. It can only optimize the manner in which the algorithm converges to its best solution for a particular image. However, it may be possible to keep multiple segmentation algorithms available and let the GA itself dynamically select the appropriate algorithm based on image characteristics. Further, it is possible to define various evaluation criteria which can be automatically selected and optimized in a complete vision system. The adaptive image segmentation system may soon be able to benefit from advances in parallel computing and very large scale integration (VLSI) technology. These hardware improvements would make it possible to achieve high quality image segmentation results at near-realtime processing rates.

REFERENCES

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

[2] Barr, A., Feigenbaum, E., and Cohen, P. R. (1981)
The Handbook of Artificial Intelligence (3 vols.).
Los Altos, CA: William Kaufman, 1981.

[3] Bethke, A. D. (1981)
Genetic algorithms as function optimizers.
Ph.D. dissertation, University of Michigan, Ann Arbor, 1981.

[4] Bhanu, B. (1986)
Automatic target recognition: State of the art survey.
IEEE Transactions on Aerospace and Electronic Systems, **AES-22**, 4 (July 1986), 364–379.

[5] Bhanu, B., Lee, S., and Ming, J. C. (1990)
Self-optimizing control system for adaptive image segmentation.
In *Proceedings of DARPA Image Understanding Workshop*, Pittsburgh, PA, (Sept. 1990), 583–596.

[6] Bhanu, B., Lee, S., and Ming, J. C. (1991)
Self-optimizing image segmentation system using a genetic algorithm.
In *Proceedings of Fourth International Conference on Genetic Algorithms*, San Diego, CA, (July 1991), 362–369.

[7] Bhanu, B., Lee, S., and Ming, J. C. (1994)
Adaptive image segmentation using a genetic algorithm.
IEEE Transactions on Systems, Man and Cybernetics, 1995.

[8] Bosworth, J., Foo, N., and Zeigler, B. P. (1972)
Comparison of genetic algorithms with conjugate gradient methods.
Washington, DC: National Aeronautics and Space Administration, 1972.

[9] Cavicchio, D. J. (1970)
Adaptive search using simulated annealing.
Ph.D. dissertation, University of Michigan, Ann Arbor, 1970.

[10] DeJong, K. A. (1975)
An analysis of the behavior of a class of genetic adaptive systems.
Ph.D. dissertation, University of Michigan, Ann Arbor, 1975.

[11] Fu, K. S., and Mui, J. K. (1981)
A survey on image segmentation.
Pattern Recognition, **13** (1981), 3–16.

[12] Goldberg, D. E. (1983)
Computer-aided gas pipeline operation using genetic algorithms and rule learning.
Ph.D. dissertation, Dept. of Civil Engineering, University of Michigan, Ann Arbor, 1983.

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

[14] Haralick, R. M., and Shapiro, L. G. (1985)
Image segmentation techniques.
Computer Vision, Graphics, and Image Processing, **29** (1985), 100–132.

[15] Holland, J. H. (1975)
Adaptation in Natural and Artificial Systems.
Ann Arbor, MI: University of Michigan Press, 1975.

[16] Holland, J. H. (1986)
Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. II.
Morgan Kaufmann Publishers, Inc., 1986.

[17] Laws, K. I. (1982)
The Phoenix image segmentation system: Description and evaluation.
Technical note 289, *SRI International*, Dec. 1982.

[18] Milgram, D. L. (1979)
Region extraction using convergent series.
Computer Graphics and Image Processing, **11** (1979), 1–12.

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

[20] Schaffer, J. D. (1985)
Multiple objective optimization with vector evaluated genetic algorithms.
In *Proceedings of International Conference on Genetic Algorithms and Their Applications*, (1985), 93–100.

[21] Schaffer, J. D., and Grefenstette, J. J. (1985)
Multi-objective learning via genetic algorithms.
In *Proceedings of 9th International Joint Conference on Artificial Intelligence*, (1985), 593–595.

[22] Selfridge, P. G. (1982)
Reasoning about success and failure in aerial image understanding.
Ph.D. dissertation, Department of Computer Science, University of Rochester, 1982.

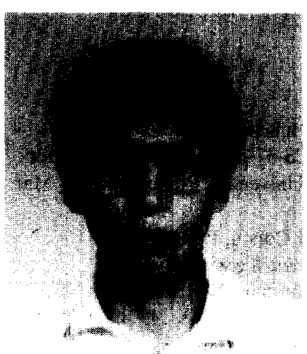
[23] Shafer, S., and Kanade, T. (1982)
Recursive region segmentation by analysis of histograms.
In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (1982), 1166–1171.

[24] Steuer, R. E. (1986)
Multiple Criteria Optimization: Theory, Computation, and Application.
New York: Wiley, 1986.

[25] Weszka, J. S. (1978)
A survey of threshold selection techniques.
Computer Graphics and Image Processing, **7** (1978), 259–265.

[26] Winston, P. H. (1984)
Artificial Intelligence.
Reading, MA: Addison-Wesley, 1984.

[27] Zeigler, B. P., Bosworth, J. L., and Bethke, A. D. (1973)
Noisy function optimization by genetic algorithms.
Technical report 143, Dept. of Computer and Communication Sciences, University of Michigan, Ann Arbor, 1973.



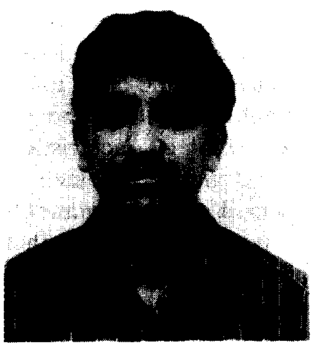
Sungkee Lee received the B.S. and M.S. degrees in electrical engineering from Seoul National University in South Korea in 1979 and 1981, respectively. He received the Ph.D. degree in computer science from the University of Utah, Salt Lake City, in 1992.

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

Bir Bhanu (S'72—M'82—SM'87) received the B.S. degree (with Honors) in electronics engg. from Institute of Technology, BHU, Varanasi, India, the M.E. degree (with Distinction) in electronics engg. from Birla Institute of Technology and Science, Pilani, India; the S.M. and E.E. degrees in electrical engg. and computer science from Massachusetts Institute of Technology, Cambridge, the Ph.D. degree in electrical engg. 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 Dr. Bhanu was a Senior Honeywell Fellow at Honeywell Systems and Research Center in Minneapolis. He has also worked with the University of Utah, Ford Aerospace and Communications Corporation, INRIA-France and IBM San Jose Research Laboratory. He has been the principal investigator of various programs from ARPA, NASA, 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.

Dr. Bhanu has 5 U.S. and international patents in computer vision and over 150 reviewed technical publications in the areas of computer vision, image processing, robotics, artificial intelligence and learning. He is the guest editor or co-editor of special issue of *IEEE Computer* on "CAD-Based Robot Vision", *Journal of Robotic Systems* on "Passive Ranging for Robotic Systems," *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 Applications* on "Innovative Applications of Computer Vision." He is on the editorial board of the *Journal of Mathematical Imaging and Vision*, the *Journal of Pattern Recognition*, and *International Journal of Machine Vision and Applications.* He is the co-author of books on *Computational Learning for Adaptive Computer Vision* (Plenum, 1995, (planned)), *Genetic Learning for Adaptive Image Segmentation* (Kluwer Academic, 1994), and *Qualitative Motion Understanding* (Kluwer Academic, 1992). He has given national short courses on Intelligent Automatic Target Recognition and he is a reviewer to over 30 technical publications and government agencies. 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 listed in the *American Men and Women of Science*, *Who's Who in America* and *Who's Who in the World*. He is a member of ACM, AAAI, Sigma Xi, Pattern Recognition Society, and SPIE.



Subhodev Das (S'82—M'92) received his B. Tech. (Hons.) degree in electronics and electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 1984, his M.S. degree in electrical engineering from the University of Hawaii, Honolulu, in 1986, and his Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign in 1991.

Between 1991–1994, he was with the College of Engineering at the University of California, Riverside. Currently, he is a research staff member at PEB, Inc., Princeton, NJ, working in real-time vision systems. His research interests include intelligent systems, computer vision, machine learning, human-computer interaction, real-time computing, parallel and distributed processing, and application of artificial intelligence, signal and image processing.