ADAPTIVE IMAGE SEGMENTATION USING A GENETIC ALGORITHM

Bir Bhanu, Sungkee Lee, and John Ming
Honeywell Systems and Research Center
3660 Technology Drive
Minneapolis, MN 55418

ABSTRACT

Machine learning will play a critical role in future computer vision systems which must operate in dynamic outdoor scenarios. In this paper, we present a system which applies a machine learning technique to the problem of image segmentation. The learning technique, known as a genetic algorithm, allows the segmentation process to adapt to changes in image characteristics caused by variable environmental conditions such as time of day, time of year, clouds, rain, etc. The genetic algorithm efficiently searches the enormous hyperspace of segmentation parameter combinations using a collection of search points known as a population. By combining high performance members of the current population to produce better parameter combinations, the genetic algorithm is able to locate the parameter set which maximizes the segmentation quality criteria. We present some initial results which demonstrate the ability to adapt the segmentation parameters to variable lighting conditions (intensity and position of light sources).

1. INTRODUCTION

An innovative combination of techniques from two branches of science can often produce significant breakthroughs in the evolution of a technology. We have developed a system which applies a technique from the machine learning field to the computer vision problem of image segmentation. Image segmentation is typically the first, and most difficult, task of any automated image understanding process. All subsequent interpretation tasks, including feature extraction, object detection, and object recognition, rely heavily on the quality of the segmentation process. Despite the large number of segmentation techniques presently available,2,8,14 no general methods have been found which perform adequately across a diverse set of imagery. Only after numerous modifications to an algorithm’s control parameter set can any current method be used to process the wide diversity of images encountered in dynamic outdoor applications such as the operation of an autonomous robotic land/air vehicle, 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. The image segmentation problem can be characterized by several factors which make parameter selection process very difficult. First, most of the powerful segmentation techniques available today contain numerous control parameters which must be adjusted to obtain peak performance. 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 non-linear fashion, which makes efficient or impossible to model their behavior in an algorithmic or rule-based fashion. Thus, the resulting objective function which results from various parameter combinations cannot generally be modeled in a mathematical way. Next, since variations between images cause changes in the segmentation results, the objective function 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 subject to 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, we need to apply a technique which can efficiently search the complex space of plausible parameter settings and locate the values which yield optimal results, without having to rely on knowledge of the particular application domain or detailed knowledge pertinent to the selected segmentation algorithm. Genetic algorithms, which are designed to efficiently locate the approximate global maxima in a search space, show great promise in solving this parameter selection problem.

The next section of this paper discusses the selection of the genetic algorithm as the appropriate search technique for this problem domain. Section 3 presents a brief overview of the details of the genetic process, including previous applications in computer vision research. Following this review, Section 4 describes the 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 5 provides some initial experimental results using the adaptive segmentation system. Finally, Section 6 discusses the conclusions of this work and describes future research plans.
2. SELECTION OF AN OPTIMIZATION TECHNIQUE

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. Figure 1 provides a generalized representation of an objective function that is typical for the image segmentation process. The figure depicts a simplified 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 can not be described in closed form.

The conclusion which can be drawn from Figure 1 is that we must identify 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 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 other complexities 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* searches have no available measure of goal distance. Branch and bound requires too many search points while dynamic programming suffers from the curse of dimensionality.
- **Knowledge-Based Techniques (Production rule systems, Heuristic methods)** - These systems have a limited domain of rule applicability, tend to be brittle, and are usually difficult to formulate. Further, the visual knowledge required by these systems may not be representable in knowledge-based formats.

*Figure 1:* Representation of the objective function which must be optimized in the adaptive image segmentation problem.
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 for any given parameter combination. Genetic algorithms are also related to simulated annealing where, although random processes are also applied, the search method should not be considered directionless. In the image processing domain, Geman and Geman have used simulated annealing to perform image restoration and Sontag and Sussmann have performed image restoration and segmentation. Simulated annealing and other hybrid techniques have the potential for improved performance over the earlier optimization techniques. However, in this paper, we will describe our experiments using genetic algorithms alone.

3. OVERVIEW OF GENETIC ALGORITHMS

Genetic algorithms were pioneered at the University of Michigan by John Holland. 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 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 can be viewed as providers of "building blocks" from which new, higher strength offspring can be constructed.

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 effect 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 search effectively minimizes the problem of converging to local maxima.

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. 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 in the genetic operations. Holland 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 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.

Since they 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 system 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 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 global maxima.

To date, genetic algorithms have been applied to a wide diversity of problems. They have been used in combinatorial optimization, VLSI layout, gas pipeline operations, and machine learning. With regards to computer vision applications, Fitzpatrick et. al 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 set of images. Genetic algorithms have also been used in computer vision for generating image domain feature detectors. Gillies.
4. ADAPTIVE IMAGE SEGMENTATION

Genetic algorithms can be used in three different fashions to facilitate an adaptive behavior within a computer system. The simplest approach is to allow the genetic system to modify the set of control parameters which affect the output of an existing computer 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. 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 allows the genetic system to actually make changes in the executable code of a program. In most cases, this adaptation involves changing the condition/action statements of a rule in a production system. Since almost every segmentation algorithm contains parameters which are used to control the segmentation results, we have adopted the first strategy listed above.

Adaptive image segmentation requires this ability to modify control parameters in order to respond to changes which occur in the image as a result of varying environmental conditions. The block diagram of our approach to adaptive image segmentation is shown in Figure 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 genetic learning component. Using this data, the genetic 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 using a set of control parameters. The details of each component in this procedure will be described in the following subsections.

4.1 IMAGE CHARACTERISTICS

The input image must be analyzed so that a set of features can be extracted to aid in the parameter selection process 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 an appropriate starting point for the parameter adaptation process.

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

- **First Order Properties**: Measure the shape of the first-order image histogram. Information includes mean, variance, skewness, kurtosis, energy, and entropy.
- **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.

\[ \text{Input Image} \]
\[ \text{Image Analysis} \rightarrow \text{Image Statistics} \]
\[ \text{Genetic Learning System} \rightarrow \text{Control Parameters} \]
\[ \text{Segmented Image Evaluation} \rightarrow \text{Segmented Image} \]

\[ \text{External Variables (Time of day, time of year, rain, snow, haze, cloud cover, etc)} \]

\[ \text{"Reward"} \]

\[ \text{Figure 2: Block diagram of the adaptive image segmentation process.} \]
• **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, and maximum peak-to-valley ratio.

For the purposes of our initial research, we have ignored the second order histogram properties since they are expensive to compute. Further, they may not be necessary if the selected first order properties and the peak/valley properties are sufficient to represent the internal image characteristics.

External variables can also be used to characterize an input image. These factors specify the conditions in 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.

### 4.2 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 simple classifier system is used to represent the image characteristics and the associated segmentation parameters. Figure 3 shows a simplified example of a classifier used by the genetic learning component. The classifier stores the current fitness of the parameter settings, the image statistics and external variables of the image, and the segmentation parameter set which is being adapted by the genetic algorithm. The image statistics and external variables form the condition portion of the classifier, $C_i$ through $C_{i+j}$, while the segmentation parameters indicate the actions, $A_i$ through $A_N$, of the classifier. Note that only the fitness value and the action portion of the classifier are subject to genetic adaptation; the conditions remain static for the life of the classifier.

The classifier used in our initial experiments is somewhat more complex than the one pictured in Figure 3 since we are using color imagery. We compute the first order histogram statistics and the histogram peak/valley properties for each of the red, green, and blue components of the color image. All of this information is then stored in the classifier. The external image variables, however, retain the same representation as shown in Figure 3.

Using the image characteristics for a new image, the genetic learning system compares this information with the current population of classifiers. The algorithm computes a ranked list of individuals which have characteristics similar to the current image. Using the highest ranked individual first, the genetic algorithm sends the parameter set from the selected individual to the segmentation component. After the image has been segmented, the results are reviewed by the evaluation system. If the segmentation quality is below a predefined threshold of acceptance, the process terminates and a new classifier is created using the characteristic values obtained from the new image, the parameter values which led to the acceptable results, and the fitness value that was achieved. The new classifier is added to the current population, replacing the individual which currently has the weakest fitness value.

![Figure 3](image)

*Figure 3: Representation of a classifier used by the genetic learning system.*
Alternatively, if after testing some preset number of the ranked classifiers, the system has not achieved acceptable segmentation quality, the genetic learning process is invoked on the set of tested classifiers. A new seed population of classifiers is temporarily created using the characteristics from the current image in each of the condition fields and the parameters sets from each of the tested classifiers in the action field. The genetic process is then applied to the seed population until acceptable performance levels are achieved or a maximum number of segmentations have been performed. At this point, some portion from the set of new classifiers is allowed to replace the weakest elements in the initial population. We only allow a fraction of the new classifiers back into the original population to avoid skewing the population towards the characteristics of the most recently processed image. Once the least fit members of the population have been replaced, the system is ready to process a new image.

4.3 SEGMENTATION ALGORITHM

Since we will be working with color imagery in our experiments, we have selected the PHOENIX segmentation algorithm developed by Shafer and Kanade at Carnegie-Mellon University. This system employs a region splitting technique which uses information from the histograms of the red, green, and blue image components simultaneously. Basically, the algorithm recursively splits regions in the image into smaller subregions on the basis of a peak/valley analysis of the various color histograms. Fifteen different parameters are used to control the thresholds and termination conditions used within the algorithm. Of these fifteen values, we have selected the two of the most critical parameters which affect the overall results of the segmentation process, maxmin and hsmooth. Maxmin specifies the lowest acceptable peak-to-valley-height ratio 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 noise in the image. Future experiments may increase the number of selected parameters used for adaptation in order to investigate more difficult segmentation tasks.

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 target in not extracted from the background, to the other extreme in which the target is broken up into many small regions that are of little use to 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. The segmentation quality criteria are described in the next section.

4.4 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 although none have 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 which uses the weighted sum of the five different quality measures as the overall fitness for a particular parameter set. The reward that is generated from this approach is a scalar measurement of the parameter set’s utility. However, a more complex vector evaluation which provides multidimensional feedback on segmentation quality can be used. In our initial experiments, we use only a scalar measure of quality for simplicity.

The measures of segmentation quality that we have selected for this work include (weighting shown in parentheses):

1. **Edge-Border Coincidence** (.20): Measures the overlap of the region borders in the image acquired from the segmentation algorithm relative to the borders found using an edge operator. In our application, we used the Sobel operator to compute the necessary edge information.

2. **Boundary Discrepancy** (.40): Measures the total number of overlapping boundary pixels in the ground truth image and the segmented image, minus the total number of non-overlapping pixels in these two images.

3. **Pixel Misclassification** (.15): Measures the number of target pixels misclassified as background pixels and the number of background pixels misclassified as target pixels.

4. **Target Contrast** (.15): Measures the contrast between the target and the background in the segmented image, relative to the target contrast in the ground truth image.

5. **Target Overlap** (.10): Measures the area of intersection between the target region in the ground truth image and the segmented image, divided by the union of the target regions.

The last four quality measures require the availability of ground truth information which represents the ideal segmentation of the image. For these experiments, the ground truth information is acquired by interactively running the PHOENIX algorithm on the image to obtain the best overall results. The weighted sum of the five quality measures is computed once each of the individual measures is known. The fitness of the parameter set is represented by this

1048
weighted sum value. In actual applications, the segmentation quality would be provided by a higher level process that was performing region labeling or object recognition and would be able to relate the quality of the segmentation with the region labeling results.

5. INITIAL EXPERIMENTS

The experiments we have performed to date use simplified scenes in which the position and intensity of the lighting have been changed. The variations in each scene are meant to approximate lighting conditions in outdoor imagery, where the angle of the sun varies over time and the intensity changes due to environmental conditions. We have selected three images in order to evaluate the parameter optimization capabilities of the genetic algorithm in the image segmentation domain. The images, shown in Figure 4, are color images (red, green, and blue components) which are 128 by 128 pixels in size. In these initial images, only the lighting intensity has been modified between scenes. The position of the lighting and the camera remain fixed. The car shown at the bottom of images is considered the object of interest for these experiments. The ground truth data for these images was obtained by interactively running the PHOENIX algorithm on Frame 1 (highest contrast image). The ground truth image is shown in Figure 5.

The first issue of concern for the experiments was the selection of the appropriate control parameters for the genetic algorithm itself, e.g. the population size, crossover rate, mutation rate, and maximum number of allowable

![Frame 1](image1.png) ![Frame 2](image2.png) ![Frame 3](image3.png)

*Figure 4:* Imagery used to evaluate the adaptive image segmentation process. In each image, the lighting intensity has been changed to simulate varying environmental conditions.

![Ground Truth Image](image4.png)

*Figure 5:* Ground truth data used for evaluating the images shown in Figure 4.
generations (or maximum number of segmentation cycles) of the genetic process. To properly select these values, and further, to validate the performance of genetic process for our image segmentation problem, we exhaustively defined the objective function for the first image (Frame 1). The two segmentation parameters that were selected for adaptation (maxmin and hsmooth) were constrained to a useful range of values and then allowed to take on 32 distinct values within those ranges. Maxmin values, which affect the segmentation quality in a non-linear fashion, were sampled exponentially over a range of values from 100 to 1100. Values near 100 were spaced closer together than values at the upper end of the range. Hsmooth values were sampled linearly, using odd numbers between 1 and 63, inclusive. This allocation provided a search space of 1024 different parameter combinations. For each parameter set, Frame 1 (Figure 4) was segmented and the results were evaluated using the quality measures described in section 4.4. The individual surfaces, along with the combined segmentation quality measure, are shown in Figure 6. Notice that the individual surfaces as well as the combined surface are very complex and can not be effectively optimized using traditional optimization techniques.

Once the surface was exhaustively defined, we were able to apply the genetic algorithm to this search space and efficiently test various combinations of genetic parameters without incurring the cost of image segmentation at every step. Additionally, since we know the maximum value of the surface, we can determine when the adaptive system has created individuals within a given threshold of the maximum. This information provides us with the stopping criterion for the first image. After 30 trial runs using various combinations of genetic parameters, we found that a population size of 10, a crossover rate of 0.8, and a mutation rate of 0.01 produced the fastest convergence rate. This set of genetic parameters allowed the adaptive system, which started from a random selection of individuals, to locate a parameter set that was within 1% of the global maximum value in only three generations (26 total segmentations). Figure 7(a) illustrates the ten randomly generated parameter locations (blackened squares) for Frame 1 at the beginning of the genetic process. Figure 7(b) displays the ten final population locations at the end of the third generation. Note that in Figures 7(a) and 7(b), some population members are not visible due to the viewing angle of the surface. In order to judge the progress of the segmentation quality at each generation, Figure 8 shows the segmentation results for the best (maximum fitness) parameter set at the end of each generation. The sequence show the increase in segmentation accuracy, as compared to the ground truth image in Figure 5, at the end of each generation. By the end of the third generation, the maximum fitness is high enough to halt any further segmentation. The ten members of the population at the end of the third generation are stored for later use in processing images with similar characteristics.

The knowledge acquired in processing Frame 1 was used during the processing of the Frame 2 and Frame 3 shown in Figure 4. Using the same population size, crossover rate, and mutation rate, the images were processed. However, in these two images, the objective function was not exhaustively computed so each individual in the population of each generation resulted in a segmentation cycle followed by a subsequent performance evaluation step. Further, since the maximum value of the objective function was not known, the stopping criteria was selected by stopping after a given number of generations, as determined from evaluating the convergence rate of Frame 1. Because of variations in the shape and maximum value of the objective function, which are caused by the minor differences in all three images in Figure 4, the number of generations before stopping was increased from three to five for Frames 2 and 3 to insure that the final results would be of high quality. Figures 9 and 10 display the best segmentation results obtained at the end of each generation for Frames 2 and 3, respectively.

In order to summarize the performance of the genetic algorithm on the objective functions for the three images, Figure 11 charts the maximum and average segmentation performance through 12 complete generations. The adaptive process was allowed to run 100 segmentation iterations to analyze the convergence rate in each of the three images. As the maximum performance chart (Figure 11(a)) shows, the genetic optimization technique achieved maximum results within 5 generations in all cases. In addition, the average performance chart (Figure 11(b)) indicates an upward trend in the performance level of the population members.

6. CONCLUSIONS

We have shown the ability of genetic algorithms to provide high quality segmentation results in a minimal number of segmentation cycles. The knowledge gained during the processing of an image can be stored for later use in a large population of classifiers which can suggest high quality classifiers for images with similar characteristics, thus avoiding the use of random parameter setting during the initial genetic processing. In some cases, the parameter settings suggested by the initial classifiers may produce acceptable segmentation results after only one segmentation cycle.

The next series of experiments currently planned will utilize the final populations created from each of the three images as a larger population from which the genetic learning system can select. As each image is sequentially processed, the new classifiers which are created will replace the weakest members of the current population and the diversity of the classifiers will begin to increase. In this manner, the system will be able to learn from experience and apply this knowledge in succeeding image processing stages.

In addition to implementing the complete genetic learning system just described, we will investigate the adaptation of additional parameter values and measure the improvement in segmentation performance by doing so. There may exist a maximum number of useful parameters, beyond which the cost of adaptation exceeds the overall

1050
Figure 6: Segmentation quality results obtained for Frame 1 in Figure 4 using various parameter combinations. (a) Edge-Border Coincidence. (b) Boundary Discrepancy. (c) Pixel Misclassification. (d) Target Contrast. (e) Target Overlap. (f) Combined segmentation quality.
Figure 7: Search points used during genetic adaptation on Frame 1, indicated as darkened squares on the objective function. (a) Randomly selected starting point locations. (b) Locations of population members at the end of the third generation, which show an increase in overall population fitness.

Figure 8: Best segmentation results of Frame 1 at the end of each generation.
Figure 9: Best segmentation results of Frame 2 (Figure 4) at the end of each generation.

Figure 10: Best segmentation results of Frame 3 (Figure 4) at the end of each generation.
improvement in segmentation quality. Further research is necessary to evaluate this hypothesis. We also plan to test the adaptive segmentation process on outdoor imagery, in which realistic lighting and other environmental changes can be observed. Using this imagery, we will evaluate the improvement in performance (effectiveness and efficiency) of our adaptive system versus one with no adaptive capabilities.

Several important findings are worth mentioning at this point. First, one crucial advantage is that the genetic algorithm can be easily applied to any segmentation technique which can be controlled through parameter changes. In addition, we can choose to adapt the entire parameter set or just a few of the critical parameters, depending on the final quality of the results that are desired. Second, it is useful to note that the adaptive segmentation system is only as robust as the segmentation technique which 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 genetic process itself dynamically select the appropriate algorithm based on image characteristics. Another important point is that, although we have only used color images in these experiments, the technique itself is applicable to any type of imagery whose characteristics can properly be represented. This set includes FLIR, LADAR, MMW, and gray scale imagery. Finally, the genetic process described in this paper may soon be able to benefit from advances in parallel computing and VLSI technology, which are now beginning to produce chips that can perform the image segmentation process in real time.17

REFERENCES


Image Understanding Workshop

Proceedings of a Workshop
Held at
Palo Alto, California

May 23-26, 1989

Sponsored by:
Defense Advanced Research Projects Agency
Information Science and Technology Office

This document contains copies of reports prepared for the DARPA Image Understanding Workshop. Included are Principal Investigator reports and technical results from both the basic and strategic computing programs within DARPA/ISTO sponsored projects and certain technical reports from selected scientists from other organizations.

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.