AD-A268 737

WL-TR-93-7017

Optimal Design of Binary Phase-Only Filters Using Genetic Algorithms

Dr. Kalyanmoy Deb

Department of General Engineering
University of Illinois at Urbana Champaign
Urbana IL 61801-2996

AUGUST 1993

FINAL REPORT FOR PERIOD FEBRUARY 1992 - FEBRUARY 1993

Approved for public release; distribution is unlimited.

WRIGHT LABORATORY, ARMAMENT DIRECTORATE
Air Force Materiel Command ■ United States Air Force ■ Eglin Air Force Base

NOTICE

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Michael L DeLorenzo

MICHAEL L. DE LORENZO, Lt Col, USAF
Chief, Advanced Guidance Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1993	Final, February 1992 - February 1993

4. TITLE AND SUBTITLE
Optimal Design of Binary Phase-Only Filters Using Genetic Algorithms

5. FUNDING NUMBERS
C: F08630-92-K-0017
PE: 61101F
PR: ILIR
TA: 92
WU: 06

6. AUTHOR(S)
Dr. Kalyanmoy Deb

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of General Engineering
University of Illinois at Urbana Champaign
Urbana IL 61801-2996

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Wright Laboratory, Armament Directorate
Advanced Guidance Division
Sensor Technology Branch (WL/MNGS)
Eglin AFB FL 32542-6810

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-93-7017

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(Maximum 200 words)*
The genetic algorithm is a mathematical optimization technique, which has generally been applied to one-dimensional problems. In this work, the genetic algorithm was applied to a two-dimensional problem--the construction of binary phase-only spatial filters for optical pattern recognition. Spatial filters that are invariant to range and aspect changes are required for robust pattern recognition. Construction of invariant filters is an optimization problem where the correlation is the objective function for the genetic algorithm. Results are presented for correlation of a genetic algorithm-constructed filter with a multiple aspect angle target set. Filters using a hill-climber algorithm were also constructed and tested.

14. SUBJECT TERMS
Optical Pattern Recognition, Optical Spatial Filters, Genetic Algorithms

15. NUMBER OF PAGES
55

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

PREFACE

This work was accomplished in the Department of General Engineering at the University of Illinois at Urbana Champaign, Urbana IL 61801-2996. The principal investigator was Dr. Kalyanmoy Deb. Drs. Dennis H. Goldstein and Thomas E. Davis, WL/MNGS, monitored the technical progress. The work was performed with Laboratory Director's Funds under Contract No. F08630-92-K-0017, and the period of performance was from February 1992 to February 1993.

The author wishes to thank Dennis Goldstein and Thomas Davis for introducing the optical filter design problem and helping clarify a number of aspects of optical filter design. Thanks is also extended to David Goldberg, a mentor of genetic algorithms, for suggestions and discussions relating to genetic algorithms and allowing use of the Illinois Genetic Algorithms Laboratory to perform this study. Karl Hadler was helpful in assisting in the performance of some of the simulations.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

List of Figures

LIST OF FIGURES (Continued)

LIST OF FIGURES (Concluded)

Chapter 1

Introduction

Optical correlators are being found to be very useful in target recognition and discrimination. Among various correlator designs, those which use discrete phase-only filters are becoming very popular because they are easy to manufacture and can have high recognition and discrimination power. The problem of designing an optimal filter is nothing more than an optimization problem. Even though there exist a number of deterministic methods to design optical phase-only filters, the researchers have suspected that solutions obtained from those methods are oftentimes local solutions (Kallman, 1990). This study investigates the application of a population-based, stochastic search and optimization technique motivated by the natural principles called *genetic algorithms* (GAs) to binary optical phase-only filter design. GAs were invented by John Holland in 1965, and since then they have been applied to a number of different problem domains including sciences, engineering, and business (Goldberg, 1989). Because of GA's *implicit parallelism* and population approach, GAs are likely to find global solutions quickly. There exist a number of other works (Calloway, 1991; Kim & Guest, 1989, 1990; Mahlab & Shamir, 1991a, 1991b, 1992) that used stochastic search methods like genetic algorithms and simulated annealing in optical filter design; but this study applies GAs in a more systematic way to a real-world military tank recognition and discrimination problem and compares the filters obtained using GAs with that obtained using an existing deterministic method and using a hillclimbing method. The images used in this study are all 128×128, requiring a total of 16,384 binary decision variables. This application is certainly one of the very largest-scale applications of GAs in real-world problems.

In the remainder of this study, a formulation of the binary phase-only filter design problem is presented. In chapter 3, a brief description of the genetic algorithms and their operators are outlined. In chapter 4, a hillclimbing method used as another optimizer is described. Chapter 5 presents and describes simulation results on a single image recognition and on multiple image recognition and discrimination problems. Chapter 6 suggests a number of avenues for further research. A conclusion is drawn in chapter 7.

Chapter 2

Binary Phase-only Filters

In this section, the principle of an optical correlator is outlined followed by the formulation of a binary phase-only filter.

2.1 Optical phase-only filters

Optical phase-only filters (as implemented in optical correlators) are used to recognize or discriminate a set of images. A schematic of an optical correlator is shown in figure 2.1. The image is first sent through a collimator lens (C). Thereafter the image is sent through a Fourier transforming lens (FFT1) and then passed through a desired filter (F) that has an effect of changing the phase of the incoming images at desired locations. The light is then sent through another Fourier transforming lens (FFT2) and finally focussed into a correlator plane (P). If a bright spot in a fairly dark background appears on the correlator plane, the filter is said to have recognized the image. Instead, if only a dark background appears in the correlator plane, the filter is said to have rejected the image.

2.2 Images

Traditionally, magnitude-only images are used to design optical filters (Bartelt & Horner, 1985; Horner & Bartlet, 1985; Hester & Casasent, 1980; Horner, & Gianino, 1984a, 1984b; Kallman, 1986a, 1986b). Only recently, Kallman and Goldstein (1991) have shown that phase-only images produce better signal-to-noise ratio compared to magnitude-only images. In this study, we have considered only phase-only images.

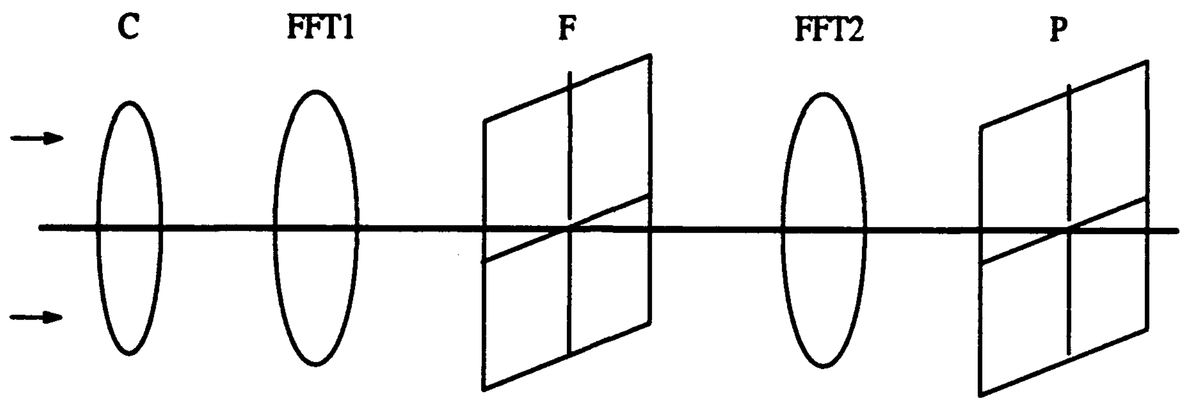

Figure 2.1 A schematic of an optical correlator.

When a detector receives an intensity image $[a_{ij}^2]$ on its active pixels, there is no reason why a mapping $[a_{ij}^2] \rightarrow [b_{ij}]$ can not be used to design a filter. In this study, we digitize the intensity into 256 different phases and use the information as unit-magnitude phase-only information. Specifically, we use the mapping $b_{ij} = \exp(a_{ij}^2 \pi \iota / 255)$. This mapping causes each pixel to have a unit magnitude and a phase in the range zero to π. This mapping is one-to-one. Elsewhere (Kallman & Goldstein, 1991), a similar mapping was considered and was found to create filters with higher signal-to-noise ratio than magnitude-only filters.

2.3 Formulation of filter design

The formulation of the filter design is described in detail elsewhere (Kallman, 1990). Here, we briefly describe the procedure.

For an image f, a complex-valued, discriminant function (filter) h is so designed that the magnitude of $\langle f, h_p \rangle$ is large for an object in f near p and is small for an object anywhere else. To obtain such a filter, the following steps are usually used:

1. Input the image f
2. Use a lens to compute the Fourier transform of f, $\mathcal{F}(f)$

3. Multiply $\mathcal{F}(f)$ by the filter $\mathcal{F}(h)^*$ using a spatial light modulator

4. Use a lens to compute the inverse Fourier transform, $\mathcal{F}^{-1}(\mathcal{F}(f) \cdot \mathcal{F}(h)^*)$, of the above product

5. Use a detector to measure the intensity $| \langle f, h_p \rangle |^2$.

A discriminant function h will be called a filter if the measured intensity in step 5 is very large for places of interest and very small elsewhere.

To use these steps to design a filter numerically, let us consider that there are a total of m images, of which first $n(< m)$ are true images (images that are required to be detected) and rest (from $n+1$ to m) are false images (images that are not to be detected). In the following formulation, we require that at least one target is a true target. We also consider that B_i is a small box of pixels in the correlation plane containing the origin in the detector phase for each true image i. For a good filter, the measured optical intensities corresponding to points in this box are larger compared to that outside the box. For false images, we consider that B_i is empty. Let us also consider that R_i is the region of the correlation plane containing the detector face and B_i for all true and false images, i. The signal-to-noise ratio of a discriminant function h is defined as follows:

$$SNR(h) = \frac{\min_{i=1}^{n} \max_{p \in B_i} | \langle f, h_p \rangle |^2}{\max_{i=1}^{m} \max_{p \in R_i - B_i} | \langle f, h_p \rangle |^2}. \tag{2.1}$$

The numerator is the minimum measured intensity of light in the region B_i in all true images and the denominator is the maximum measure outside B_i in all true and false images. In order for the discriminant function h to be a filter, the above SNR value must be as large as possible. Ideally, we should have a filter for which the numerator is as high as possible, and the denominator is as small as possible. This problem can be best solved by transforming the above problem into a multicriterion optimization problem, but in this study we use a single-objective optimization problem by maximizing the above equation.

It is interesting to note that for a single image, $n = m = 1$ and the above SNR value reduced to the definition of a signal-to-clutter ratio used in radar.

Like a continuous matched filter, a continuous phase-only filter may be envisioned with each $\mathcal{F}(h)^*(p)$ being a complex valued number of modulus one. Obviously, such a filter is hard to build. The continuous filter can be approximated by building a discrete k-state phase-only filter allowing only k different possible phase changes at each pixel p. In this study, we choose $k = 2$ and design only binary phase-only filters. Thus, in a binary phase only filter, each $\mathcal{F}(h)^*(p)$ can either be a 1, -1, or a 0 depending on whether the desired phase change is π, zero, or indifferent.

Once the choice of R_i and B_i has been fixed, the number of decision variables to be used in the optimizer can be calculated. The images f_i were supplied by the Wright Laboratory Armament Directorate. In our study, the images are of size 128×128. In all our simulations, we have chosen R_i to be also the box 128×128. Thus, there are a total of 16,384 binary decision variables. An optimal choice of B_i requires some experimentation. We use a box of size 31×31 centered in the origin. This size was used by Kallman (1987). Thus, the phase-only filter design problem reduces to finding 16,384 different binary values for which the calculated SNR value is maximum. It has been found elsewhere (1986b) that making a few pixels zero near and including the origin in the filter plane produces higher SNR values. As used in Kallman (1987), we make a box of size 11×11 centered in the origin to take a value zero. Thus, the total number of decision variables reduces to $16,384 - 121$ or $16,263$.

Chapter 3

Genetic Algorithms

Genetic algorithms (GAs) are search and optimization procedures motivated by natural principles and selection (Goldberg, 1989; Holland, 1975). Darwin's survival-of-the-fittest principle along with structured recombination operators are applied iteratively on a population of strings representing the problem-variables to evolve better populations. In this section, we briefly describe the mechanics of genetic algorithms by first outlining GA operators and then briefly discussing why GAs work. Finally, a new recombination operator is designed for solving the binary phase-only filter design problem.

3.1 Representation

In order to apply GAs to an optimization problem, the decision variables are usually mapped and represented by a string (a *chromosome*) of binary alphabets (*genes*). For problems with more than one decision variable, each variable is usually represented by a substring, which are then concatenated together to form a bigger string. Even though coding decision variables in binary is mostly used, there are some studies with decision variables being coded with a higher cardinality alphabet (Grefenstette & Fitzpatrick, 1985; Antonisse, 1989). There exist some studies where the decision variables are coded in real numbers (Eshelman & Schaffer, in press; Wright, 1991).

A flow chart of the working of a simple GA is outlined in figure 3.1. A population of strings is created at random. Each string is then evaluated. The evaluation procedure first requires decoding of the decision variables from the string. Once the values of the decision variables are decoded, they are used to calculate the objective function value, which is used as a measure of the 'goodness' of the string. In GA's terminology, the objective function value of a string is loosely known as the *fitness* of the string.

3.2 Operators

In a simple genetic algorithm, there are three main operators that are used to modify a population of strings.

Reproduction (or selection) is an operator that makes more copies of better strings in a population. There are a number of reproduction schemes in the GA literature (Goldberg & Deb, 1990). But there are two that are mostly used—proportionate selection and tournament selection. In the proportionate selection scheme, a string is selected with a probability f_i/f_{avg}, where f_i is the fitness of the ith string and f_{avg} is the average fitness of all strings in the population. This indicates that a string with higher fitness value has a higher probability of getting selected than a string with comparatively lower fitness value. In a binary tournament selection scheme, two strings are selected at random and the better string is chosen. If performed without replacement, this indicates that every string in the population will be used in exactly two tournaments. The best string will win both times; so it will get two copies. The worst string will loose both times; so it will get no copies. In different selection schemes, there are fundamental differences by which the number of copies are assigned, but the essential idea of a reproduction scheme is that more copies are allocated to the string with higher fitness value.

After the reproduction phase is over, the population is enriched with good strings. Reproduction makes clones of good strings, but does not create any new string. A crossover operator is used to recombine two strings at a time to hopefully create a better string. There exists a number of crossover operators in the GA literature (Booker, in press; Spears & De Jong, 1991; Syswerda, 1989). A single-point crossover operator is mostly used. Two strings are chosen at random for a crossover. A crossing site is chosen

```
Initialization
Evaluation
repeat
    Reproduction
    Crossover
    Mutation
    Evaluation
until (termination criterion)
```

Figure 3.1 A flowchart of the working of a genetic algorithm.

at random. The contents in the left of the crossing site are swapped between the two strings. There exists a number of variations of this crossover but the essential idea is to exchange bits (information) between two good strings to desirably obtain a string that is even better than the parents.

Another operator—mutation—is used sparingly. A bit is changed from one to another at random. Under this operator, a 1 will change to a 0 and vice versa. Mutation also creates a new string, but its effect is considered to be local. It introduces diversity in the population whenever the population tends become homogeneous due to iterative use of selection and crossover operators.

After new strings are created, they are evaluated by decoding and calculating the objective function value. This completes a cycle of GA iteration. All three operators are again applied to this population to create a new and hopefully better one. These cycles (known as *generations*) continue until a termination criterion is satisfied.

3.3 Why do they work?

The selection of good strings in a population and random information exchange among good strings are very simple and easy. But how do such simple and randomized mechanisms make a successful search? Even though there is no rigorous mathematical proof of convergence of GA search yet, some answer to this question is given in the literature (Goldberg, 1989; Holland, 1975) from a schema-processing point of view. Without going into the details, it may suffice here to note that a schema represents a set of strings with similarities in certain string positions. For example, a schema 1 0 * * * represents all eight strings with having a 1 and a 0 in the first and second positions respectively. A * denotes a don't care, meaning that it can take either a 1 or a 0. Holland (1975) has argued that in one generation of a GA with n strings in the population, a total of n^3 schemata get processed in parallel. This implicit processing of such a large number of schemata allows simultaneous schema competition among a large number of schema partitions. It is in this aspect that the fundamental theorem of genetic algorithms is hypothesized—low-order schemata combine to form high-order schemata. It is also believed that in order for schema processing, the GA operators may be such that the good schemata (building blocks) grow in the population with generation. The survival and continual growth of building blocks depends on a number of factors, including the coding and genetic operators (Kargupta, Deb, & Goldberg, 1992; Radcliffe, in press; Vose, in press).

3.4 GAs in binary filter design

In order to apply GAs to the binary phase-only filter design problem, we use a representation scheme and a crossover scheme that respects the underlying building blocks in that representation.

Since only binary phases are allowed, each decision variable is a 1 or a 0 depending on whether the phase is 180 degrees or zero degrees. Since the images are of size 128×128, there is a total of 16,384 binary decision variables in a string. This is in any standard one of the largest-scale problems tried using GAs.

Parent 1	Parent 2		Child 1	Child2
1111 111111 11	0000 000000 00		1111 000000 11	0000 111111 00
1111 111111 11	0000 000000 00		1111 000000 11	0000 111111 00
1111 111111 11	0000 000000 00		1111 000000 11	0000 111111 00
1111 111111 11	0000 000000 00		0000 111111 11	1111 000000 00
1111 111111 11	0000 000000 00		0000 111111 11	1111 000000 00
1111 111111 11	0000 000000 00	$\Longrightarrow$	0000 111111 11	1111 000000 00
1111 111111 11	0000 000000 00		0000 111111 11	1111 000000 00
1111 111111 11	0000 000000 00		1000 111111 11	1111 000000 00
1111 111111 11	0000 000000 00		1111 111111 00	0000 000000 11
1111 111111 11	0000 000000 00		1111 111111 00	0000 000000 11
1111 111111 11	0000 000000 00		1111 111111 00	0000 000000 11
1111 111111 11	0000 000000 00		1111 111111 00	0000 000000 11

Figure 3.2 A two-dimensional crossover operator used in the filter design.

It is not trivial to identify building blocks in such a problem and probably it depends on the images used. But it is intuitive that building blocks constitute variables that are geometrically close to one another. It is in this spirit that we design a crossover operator that respects geometrically close variables. A similar crossover operator was used elsewhere (Callaway, 1991). First, the string is arranged to form a two-dimensional squared array of size 128. Two sites along both row and column are chosen at random. This divides the total squared array into nine different rectangular regions. In one crossover operation, each region is swapped among two mating strings with a probability 1/9. Only one region on average gets swapped between two strings. This crossover operator is demonstrated in figure 3.2.

Mutation is performed as usual. With a small probability, a 1 is changed to a 0 and vice versa. The mutation probability is set so as to alter (on an average) a certain number of bits in a string. For example, if five bits are desired to be changed on an average, the mutation probability is set to 5/16384 or 0.00031.

Chapter 4

A Hillclimbing Technique

In addition to using genetic algorithms for the design of binary phase-only filters, a hillclimbing technique is also used. There exists a number of hillclimbing algorithms for optimization (Ackley, 1987), but in this study we use a simple hillclimbing method.

The algorithm begins with a filter generated by binary approximation of a matched phase-only filter. Starting from the top-left corner of the filter moving to the right, each pixel is complemented (a `1` is changed to a `0` and vice versa). If the SNR value of the new filter is better than the old filter, the new filter replaces the old filter and the bit flipping is continued on the new filter. If, however, the SNR value of the new filter is no better than the old filter, the subsequent bit flipping is continued on the old filter. Once all pixels in the filter are considered, the bit flipping is continued from the top-left pixel again. This procedure is terminated when for all pixels no bit flipping results in an increase in SNR value. Figure 4.1 shows a pseudo-code for this algorithm.

```
Initial filter F;
repeat
    index = 1; convergence = 0; Fnew = F;
    repeat
        Fnew[index] = Complement(F[index]);
        Evaluate Fnew;
        if SNR(Fnew) > SNR(F) then
            F[index] = Fnew[index];
            convergence = convergence + 1;
        else Fnew[index] = F[index];
        index = index + 1;
    until index = 16384;
until convergence = 0;
```

Figure 4.1 A pseudo-code for the hillclimbing algorithm.

Chapter 5

Simulation Results

In this section, we discuss the simulation results of binary phase-only filter design using genetic algorithms and using the hillclimbing technique described in previous sections. First, simulation results for a single image recognition are described. Thereafter, simulation results for multiple image recognition and discrimination are described.

In all GA simulations described here, binary tournament selection is used. The two-dimensional crossover operator described in the previous section is used and a simple mutation is used. All figures are plotted with an average of three independent simulations[1], unless otherwise noted.

5.1 Single image recognition

To investigate how GAs perform on BPOF design, they are first applied to a single image recognition problem. The target image is taken as the image at zero degrees from the front of a M60 tank. This imagery was supplied by the Wright Laboratory Armament Directorate at Eglin Air Force Base, Florida. The filter to be designed consists of 128×128 binary phase-changes of zero or π. As discussed earlier, a small box of 11×11 centered at the origin is excluded from varying. Thus, a string has a total of 16,263 binary decision variables.

---

[1]In each run, a different random seed is used, whereas all GA parameters, like population size, crossover and mutation probabilities, are kept fixed.

5.1.1 Random initial population

In the first experiment, a random initial population of 100 is used. A crossover probability of 0.8 is used. To maintain diversity, a mutation probability equivalent to ten bit changes per string is used.

The best string (filter) at the initial population has a SNR value equal to 1.274 and after 100 generations (a total of 8,000 function evaluations) the string has a SNR value equal to 9.188. A filter with a SNR value equal to 9.188 is not bad, considering that in the initial population the best string has a SNR value equal to 1.274. Moreover, this result is encouraging if the number of function evaluations is compared with the total search space. With 16,263 binary decision variables, there are a total of 2^{16263} or $4.47(10^{4895})$ different filters possible. GAs only evaluated a tiny fraction of $2(10^{-4892})$ of the search space.

In order to compare this SNR value with that obtained using a continuous phase-only matched filter, we use $\mathcal{F}(h)^*$ described in Kallman (1990). For a single image f, a phase-only matched filter is designed as follows:

$$\mathcal{F}(h)^*(p) = \begin{cases} 0, & \text{if } \mathcal{F}(f)(p) = 0; \\ \mathcal{F}(f)(p)^* / \mid \mathcal{F}(f)(p) \mid, & \text{otherwise.} \end{cases} \tag{5.1}$$

This construction sets the amplitude of each pixel to one. In order to make the comparison reasonable, the same small box of size 11×11 centered at the origin is set to zero. Thus the filter is designed using equation 5.1 except for the small box centered at the origin for which $\mathcal{F}(h)^*(p) = 0$. Using this construction, a filter is designed with a SNR value as high as 272 for the same image considered above. Even though phase-only matched filters are hard to build, such a high SNR is possible. When compared to this value, the SNR obtained by the above GAs with random initial populations of size 100 is not so good. An earlier investigation reveals the following observations:

- Since the initial population is random and since the search space is large, it is very unlikely to expect any underlying building block in the population.

- A population size of only 100 is used. This population may be very small for problem with strings of size 16,263. Elsewhere (Goldberg, Deb, & Clark, 1992), a population sizing equation has been obtained for simple genetic algorithms. For a correct decision making within GAs, a population size of $O(\ell)$ has been suggested, where ℓ is the string length. This suggests that in our problem a population size of the order of ten to twenty thousand is required.

We have also observed in the binary phase-only design literature (Horner & Gianino, 1984a, 1984b, 1985; Horner & Leger, 1985; Kallman, 1987, 1988a, 1988b, 1990) that the initial filters used to drive various optimization algorithms are not random filters but filters that are discrete versions of a matched filter. Starting from a good initial point helps the search to converge to an optimal solution quickly. Moreover, if some knowledge about the problem is known it does no harm to use that information in the algorithm. Thus, we use an initial population of strings that are not random but a few random mutations of a discrete version of the matched filter. But before we use this initial population, we ran a GA with a random initial population of size 1,000 to see if a better filter is found. The best filter in the initial population has a SNR value equal to 1.821, and at the end of 100 generations the filter has a SNR value equal to 15.563. Even though other GA parameters used in this simulation are not optimal in any sense, we are somewhat convinced that the filters obtained from random initial population are inferior compared to a matched filter. In subsequent simulations, we have always used a knowledge-augmented initial population.

5.1.2 Knowledge-augmented initial population

For the single image recognition problem, the initial population is created by first making a binary version of the matched filter. If the phase of the matched filter at any pixel p is greater than zero and at most equal to π, $\mathcal{F}(h)^*(p) = 1$ is assumed, otherwise $\mathcal{F}(h)^*(p) = -1$ is set. A population is created by flipping a certain number of bits at

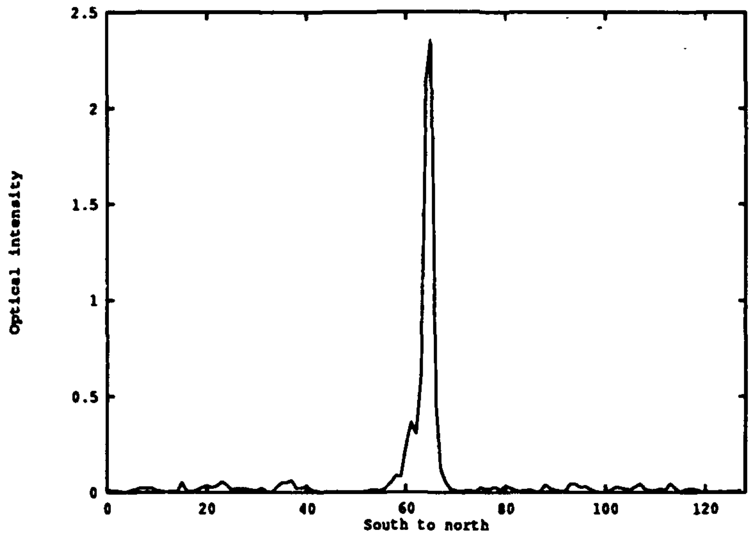

Figure 5.1 The south-to-north correlation plane view for a BPOF designed for an east-facing M60 tank using genetic algorithms. The SNR value is as high as 274.39.

random. In all our simulations, we have flipped 128 bits at random.

An initial population of good filters reduces the *fitness variance* (Goldberg, Deb, & Clark, 1992). The population sizing equation developed in that study indicates that with reduced fitness variance the required population size may reduce drastically. In this study, no effort is made to calculate the required population size in this type of problem. In all our simulations, we use a population size of 1,000 unless otherwise stated.

GAs with a population size 1,000, a crossover probability of 1.0, and a mutation of 10 mutations per string are run on this filter design problem. The best filter in the initial population has a SNR value equal to 109 and after 100 generations has a SNR value equal to 274.39. This SNR is slightly higher than that obtained by a binary phase-only matched filter. The south-to-north correlation plane view of this filter on the image is shown in figure 5.1. It is important to note that this SNR value is obtained with a reasonable set of parameter settings. A better filter may be designed with a better setting of GA parameters.

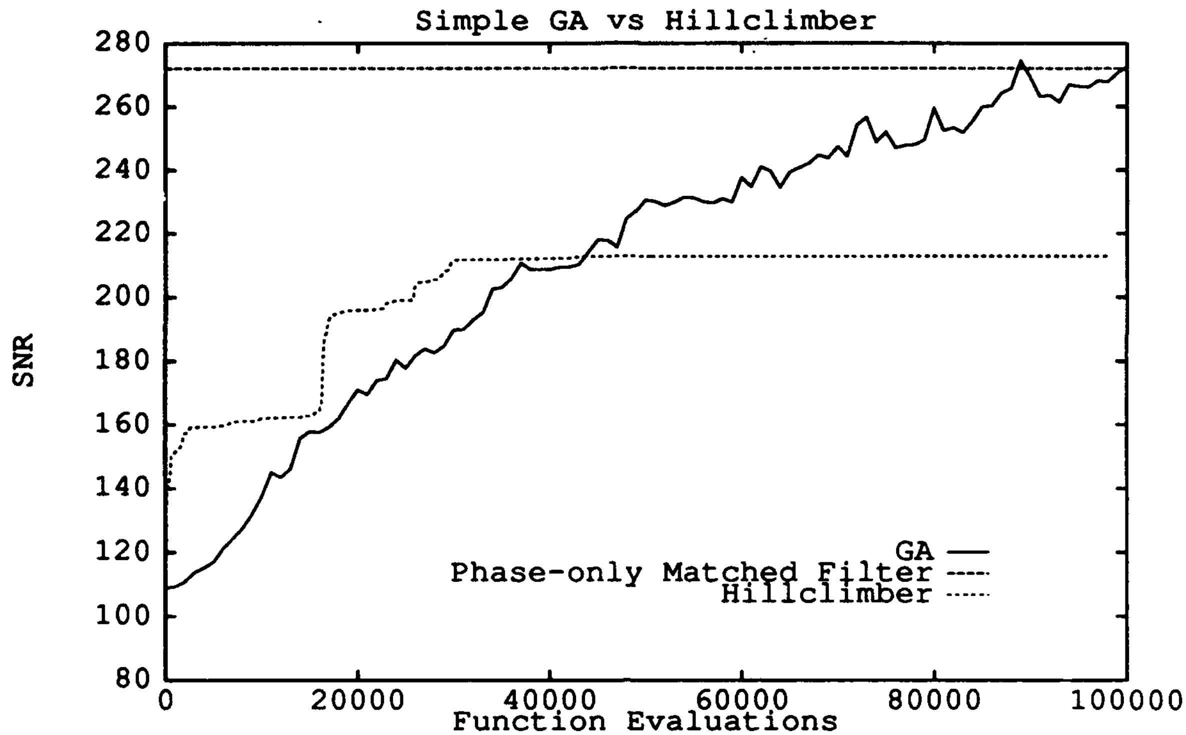

Figure 5.2 SNR values for the population-best filter in GA simulation, for the filter obtained by the hillclimber, and for the phase-only matched filter are plotted versus the number of function evaluations. The GA obtained a filter about 30% better than that obtained by the hillclimber and is better than that of the phase-only matched filter.

To investigate how well GAs compare to a local search method, the hillclimbing algorithm described in section 4 is used next. To make the comparison fair, the hillclimbing method is started with the same binary version of the phase-only matched filter used in GAs. The initial filter has a SNR value equal to 94.65 and at the end of the simulation, the filter has a SNR value equal to 212.75. This value is about 22.5% lower than that obtained by GA simulations. Figure 5.2 compares the SNR values obtained by GA simulation, by a phase-only matched filter, and by the hillclimbing method. For GA simulation, the SNR value for the population-best filter is shown. The figure shows that the GA obtained a filter better than even the binary phase-only matched filter after about 90,000 function evaluations.

These simulation results show indications that the simple GAs may be used to design better filters. As discussed earlier, these GA simulations were obtained with reasonable values of the GA parameters. In the following, we perform a parametric study to obtain better filters.

5.2 Parametric study

The important GA parameters are population size, crossover probability, mutation probability, and the selection pressure. In all our simulations, the binary tournament selection is used. Thus, the selection pressure for the best string is always two. In the following parametric study, when one parameter is varied, other parameters are held constant. In the simulations to follow, we first vary the mutation probability. Thereafter we vary crossover probability and finally we vary the population size.

5.2.1 Mutation rate

Mutation is performed by altering a certain number of bits per string. We fix the mutation probability in such a way that on average a certain number of bits are changed per string. Specifically, we vary mutation rate so that 5, 15, and 20 bits get mutated per string per generation. In all simulations, a population size of 1,000 and a crossover probability of 1.0 is used. Even though these values may not be optimal, later on we shall see that these values produce good filters. In all simulations, GAs are run for 200 generations. An average of three runs are plotted. The SNR value of the population-best string is plotted in figure 5.3 with the number of function evaluations for three different mutation rates. The figure shows that a better filter is produced with a smaller mutation rate. At the end of 200 generations, the best SNR value is found to be 310, which is about 14% better than the matched phase-only filter. In the initial few generations, GAs with all three different mutation rates perform equally well, but later

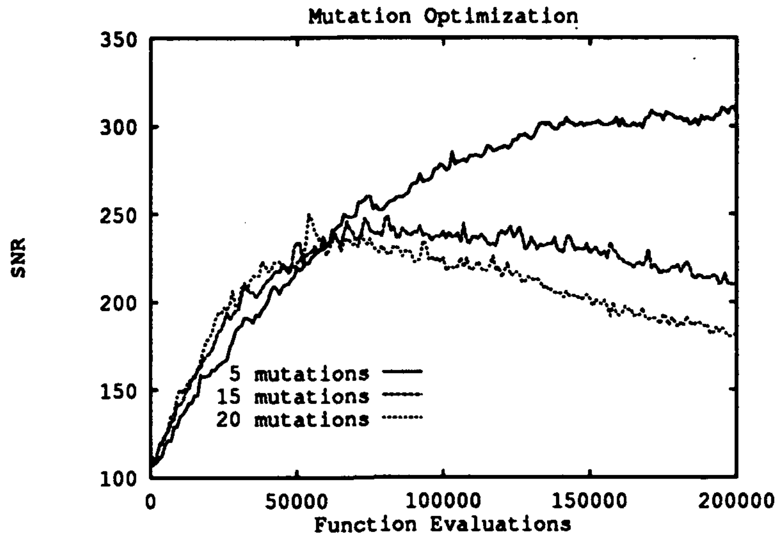

Figure 5.3 The variation of SNR value with function evaluations for different values of mutation rates. Better filters are produced with smaller mutation rates.

on GAs with smaller mutation rates produce a much better filter. In most GA studies, a large crossover rate and a small mutation rate are used. Motivated by the above results, we use five-bit mutations per string in subsequent studies.

5.2.2 Crossover rate

The two-dimensional crossover operator works by choosing two random sites both along the row and column, dividing the filter plane into nine different regions and then interchanging on average one of the regions between the two mating strings. Even though we could vary the number of regions to be interchanged, in all our simulations we held that constant. Instead we only vary the probability of crossover. Three different crossover probabilities—0.8, 0.9, and 1.0—are used. For example, with a crossover probability of 0.8, 80% of the population is used in the crossover operation. In all simulations, a population size of 1,000 and a mutation rate of five bit-mutations per string are chosen. The SNR value for the population-best filter is plotted versus function

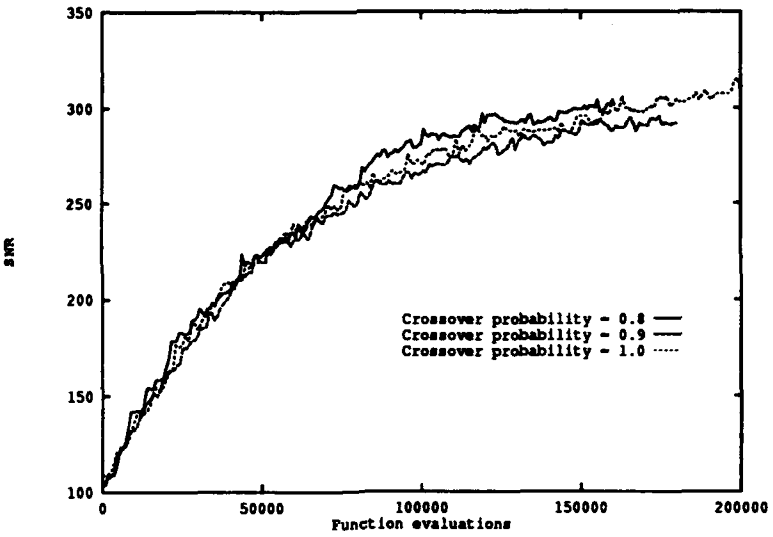

Figure 5.4 The variation of SNR value with function evaluations for different values of crossover probabilities. A high crossover probability produces a better filter.

evaluations in figure 5.4. The figure shows that GAs with crossover probabilities of 0.8 and 1.0 perform slightly better than that of 0.9. But for all practical purposes, they all perform equally well. The best filter found has a SNR value equal to 310, which is about 14% better than that of the matched filter.

5.2.3 Population size

Elsewhere (Goldberg, Deb, & Clark, 1992), a population sizing equation is developed to take into account the signal-to-noise ratio in the problem. The sizing equation suggests that for problems of bounded difficulty, a population of size $O(\ell)$ is required, where ℓ is the problem length. That equation was developed to ensure that building blocks of a certain signal-to-noise ratio would be detected with a certain probability in the initial random population. According to this sizing, we need a population size on the order of ten thousand, since our string length is 16,384. However, the population sizing can be drastically reduced if a knowledge-augmented initial population is used.

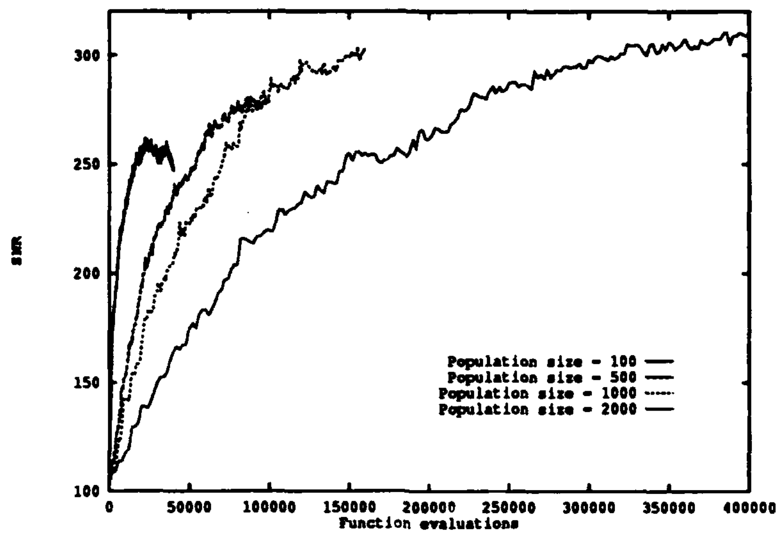

Figure 5.5 The variation of SNR values with the number of function evaluations for different population sizing.

Even though that study did not account for population sizing in such a case, the reason for reduction in population sizing is that in a knowledge-augmented initial population building blocks are associated with small noise. In this study, we have used four different population sizes—100, 500, 1000, and 2000. The crossover and mutation probabilities are held fixed at 0.8 and five bit-mutations per string, respectively. The SNR value for the population-best string is plotted in figure 5.5 versus generation number. It is clear from the figure that GAs with a small population size of 100 have not been able to find a better filter after a few thousand function evaluations. A better filter has been found with a larger population size. By comparing results for population sizes of 1000 and 2000, we find that to obtain a similar performance more function evaluations are required with a larger population size. In all our subsequent GA simulations, we use a population size of 1000.

5.3 Multiple image recognition

In the multiple image recognition problem, we use images of a M60 tank in five different orientations for recognition and simultaneously use images of a M113 tank in the same five orientations for discrimination. Three different sets of images are studied here. First, five consecutive images are tried. Thereafter, five images at 5 degrees interval are studied. Finally, nine images at 5 degrees interval are considered.

5.3.1 Consecutive images

For both M60 and M113 tanks, the images at 0, 1, 2, 3, and 4 degrees from the front of the tank are considered. We consider the images of the M60 tank to be true images, since they are used for recognition, and the images of the M113 tank to be false images, since they are used for discrimination. The evaluation of the SNR value for a binary filter is described in an earlier chapter. With five true and five false images, we substitute $n = 5$ and $m = 10$ in equation 2.1. To reduce the required population size, we use a knowledge-augmented initial population. Initial population is calculated in two different ways. In the first approach, one-fifth of the population is created by mutating a binary version of the matched phase-only filter corresponding to each true image. The initial population is created by flipping 128 bits of this binary filter at random. In the second approach, a binary filter is first created by averaging the matched phase-only filters corresponding to all five true images. The initial population is then filled with filters created by flipping 128 bits of this binary filter at random.

The hillclimber is used first. The initial filter is constructed by first calculating the average of the matched phase-only filters of all five true images and then by creating a binary filter from the average-matched phase-only filter. The SNR value of this initial filter is found to be equal to 7.503. After 278,514 iterations, the hillclimber has found a filter with a SNR value equal to 59.420. The variation of the SNR value with iteration

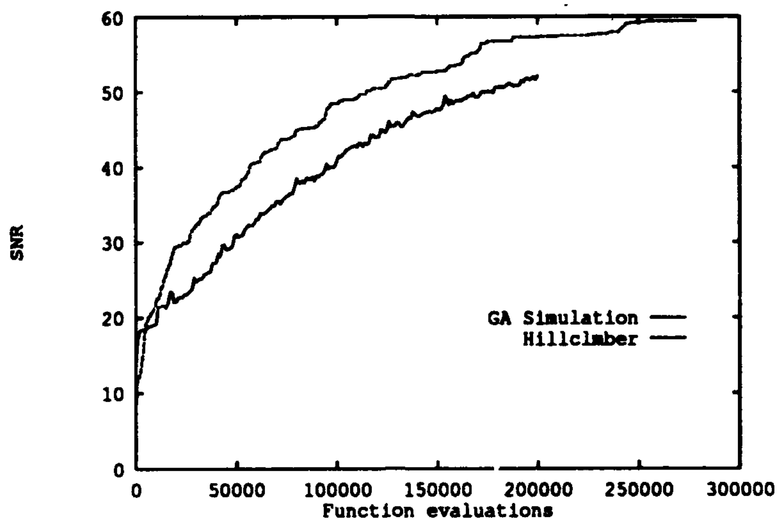

Figure 5.6 The performance of the hillclimber and genetic algorithms on five consecutive images.

number is shown in figure 5.6.

Genetic algorithms are applied next to this multiple image recognition problem. The best parameter setting found for the single-image recognition problem are used here. Even though the best parameter setting obtained for the single-image recognition problem may not be optimal for the multiple-image recognition problem, since the characteristics of the two problems are the same, it is expected that similar parameter setting would produce reasonable performance for multiple image recognition problem. GAs with a crossover probability of one and five bit mutations per filter are used. A population size of 1,000 is used in all cases. The initial population is created by randomly flipping 128 bits of the binary version of the matched phase-only filter of each of five true images. The best filter in the initial population has a SNR value equal to 12.105. The best GA run with three different initial populations found a filter at the end of 200 generations with a SNR value equal to 52.167. Figure 5.6 shows that better filters are found with more function evaluations. The SNR values obtained for M60 images at

0, 5, 10, 15, and 20 degrees are 54.044, 79.511, 102.622, 77.804, and 52.167. The SNR values obtained for the M113 images at 0, 5, 10, 15, and 20 degrees are 0.967, 0.956, 0.930, 1.042, and 0.961 only. The minimum of the maximum of $| \langle f, h_p \rangle |^2$ inside the box B obtained for five true images is found to be 5.430 and the maximum of the maximum of $| \langle f, h_p \rangle |^2$ outside B for five true images and all through R for five false images is found to be 0.104.

Even though the hillclimber finds a filter better than that obtained by GA simulation, it can be seen from figure 5.6 that the hillclimber has completely converged to the obtained filter, whereas the performance of the GA is still improving. The GA has been terminated sooner than its expected convergence time. Since no such analysis exists in the GA literature to account for the expected convergence time, the GA has been arbitrarily terminated at 200 generations. If possible, such analysis would involve knowledge about underlying building block size, string length, population size, and underlying scaling of the building blocks. In subsequent simulations, the GA has been run for more generations.

When the initial population is created by flipping the binary version of the average of the matched filters corresponding to five true images, the GA performs worse than above. Thus, in all subsequent GA simulations, the initial population is created with the former approach.

The filter obtained by using the GA has been used to find the SNR value for a few other images. Figures 5.12 to 5.14 show that the SNR value for a true image reduces as the orientation of the image varies too much from those considered in the design. For the image at 5 degrees from the front of the tank the SNR value is 32.575, for the image at 10 degrees from the front of the tank the SNR value is 6.749, and for image at 20 degrees from the front of the tank the SNR value is 2.174. For false images, these values are 1.233, 2.146, and 0.582, respectively. Figure 5.15 shows how SNR value reduces as a distant image is used for the filter generated using GAs. The obtained filter has not been

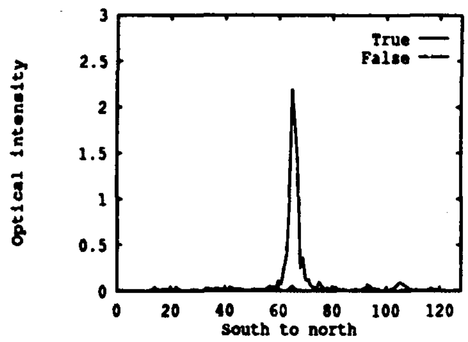

Figure 5.7 Optical intensity in the correlation plane for 0 degrees true and false image obtained for the BPOF generated by GAs.

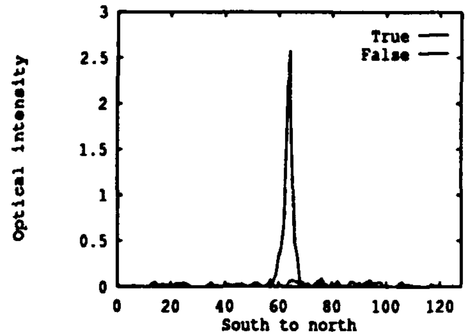

Figure 5.8 Optical intensity in the correlation plane for 1 degrees true and false image obtained for the BPOF generated by the GAs.

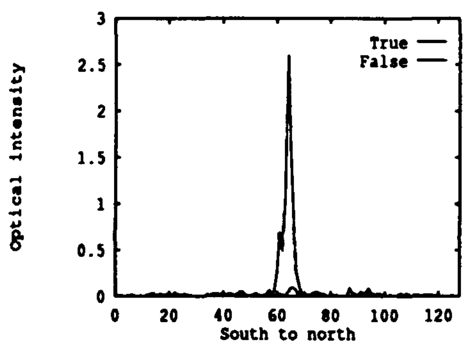

Figure 5.9 Optical intensity in the correlation plane for 2 degrees true and false image obtained for the BPOF generated by GAs.

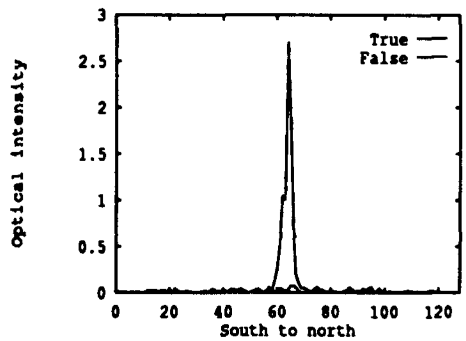

Figure 5.10 Optical intensity in the correlation plane for 3 degrees true and false image obtained for the BPOF generated by GAs.

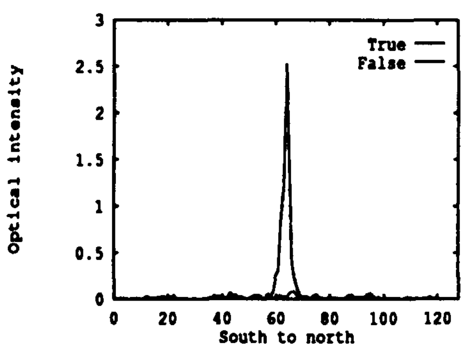

Figure 5.11 Optical intensity in the correlation plane for 4 degrees true and false image obtained for the BPOF generated by GAs.

able to discriminate true and false images that are very different than those used in the design; but for images included in the design the obtained filter is able to discriminate two tanks very well.

One of the reasons for achieving such high values of SNR is that the filter is designed for consecutive images. Images that are only one degree apart from each other are very similar. In the following subsection, we use five images that are five degrees apart from each other.

5.3.2 Distant images

The true and false images are chosen from the same tanks; but now images at 0, 5, 10, 15, and 20 degrees from the front of the tank are used. The filters obtained using the hillclimbing method and genetic algorithms are presented and compared with that obtained by Kallman's method (1990).

We obtained a binary phase-only filter for these images obtained using Kallman's method from Dennis Goldstein of the Wright Laboratory Armament Directorate (Goldstein, 1992). The SNR value for these images is found to be 27.831. The minimum of the maximum of $| \langle f, h_p \rangle |^2$ inside the box B obtained for five true images is found to be 4.067 and the maximum of the maximum of $| \langle f, h_p \rangle |^2$ outside B for five true images and all through R for five false images is found to be 0.146. We now use the hillclimber and GAs to design filters for these images.

The initial filter to the hillclimber is calculated by creating a binary filter from an average of the matched phase-only filters corresponding to the true images. The SNR value of this filter is 9.105. The filter obtained by the hillclimbing method after 81,927 function evaluations has a SNR value equal to 36.524. The obtained filter has a SNR value more than 30% better than that obtained using Kallman's method. The minimum of the maximum of $| \langle f, h_p \rangle |^2$ inside the box B obtained for five true images is found to be 5.347 and the maximum of the maximum of $| \langle f, h_p \rangle |^2$ outside B for five true images

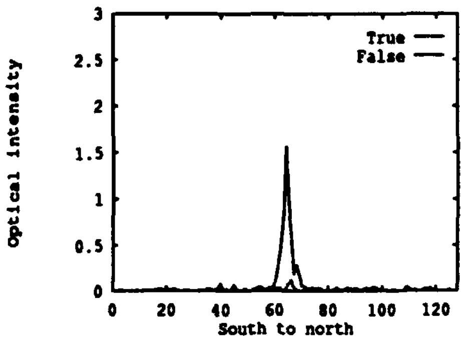

Figure 5.12 Optical intensity in the correlation plane for 5 degrees true and false images. Even though these images are not used in the design, the filter found using GAs is able to discriminate an M60 and an M113 tank.

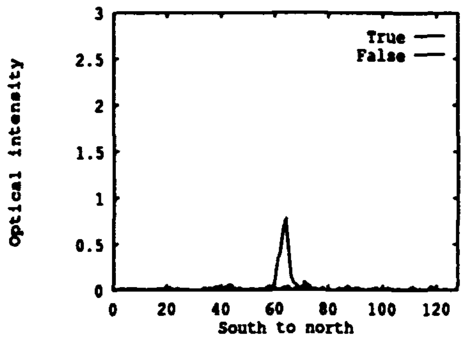

Figure 5.13 Optical intensity in the correlation plane for 10 degrees true and false images. Even though these images are not used in the design, the filter found using GAs is able to discriminate an M60 and an M113 tank.

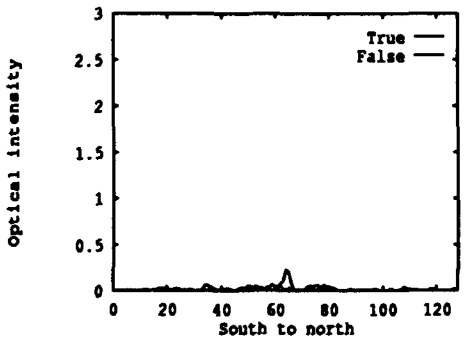

Figure 5.14 Optical intensity in the correlation plane for 20 degrees true and false images. The filter found using GAs is not able to discriminate an M60 and an M113 tank at 20 degrees from the front of the tank.

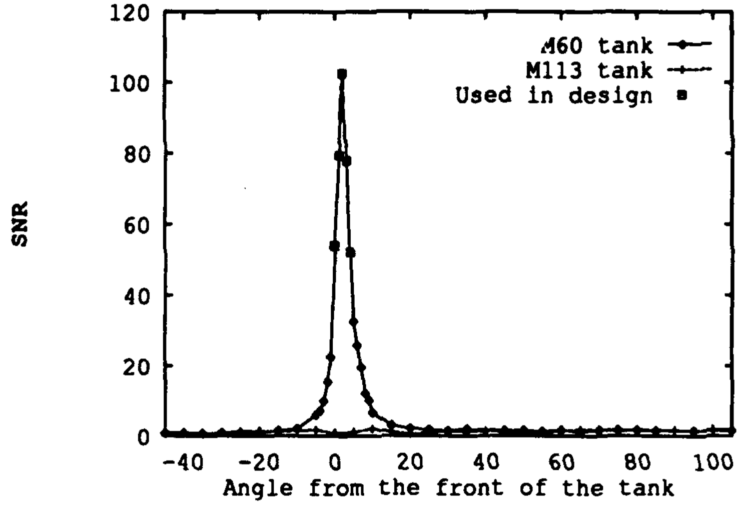

Figure 5.15 SNR values for images at different orientations with the BPOF designed by genetic algorithms.

and all through R for five false images is found to be 0.146. When each of the true and false images is used to find the SNR for this filter, they vary from 36.525 to 50.472. The optical intensities corresponding to these images are plotted in figures 5.16 to 5.20. In all cases, the obtained filter has been able to recognize the M60 tank and discriminate it from the M113 tank on these images. The SNR values obtained for M60 images at 0, 5, 10, 15, and 20 degrees are 36.526, 50.473, 46.953, 45.461, and 36.588. The SNR values obtained for the M113 images at 0, 5, 10, 15, and 20 degrees are 1.033, 0.999, 0.999, 1.056, and 1.059.

After achieving good results with the hillclimber, GAs are applied next. GAs with a population size of 500, with crossover probability of 1.0, and five mutations per filter, are used. The initial population is created by filling one-fifth of the population by flipping 128 bits of each binary filter generated from the matched phase-only filter of the true images. The best filter in the initial population has a SNR value equal to 6.158 and after 300 generations the best filter has a SNR value equal to 28.425. This filter

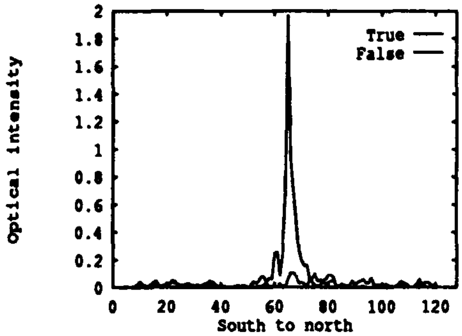

Figure 5.16 Optical intensity in the correlation plane for 0 degrees true and false image obtained for the BPOF generated by the hillclimber.

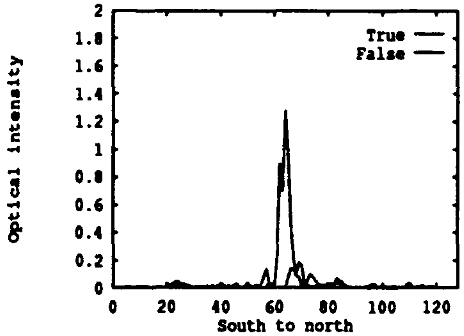

Figure 5.17 Optical intensity in the correlation plane for 5 degrees true and false image obtained for the BPOF generated by the hillclimber.

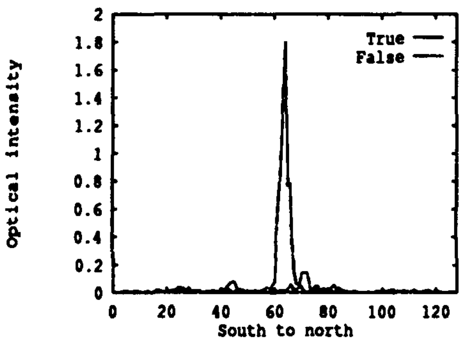

Figure 5.18 Optical intensity in the correlation plane for 10 degrees true and false image obtained for the BPOF generated by the hillclimber.

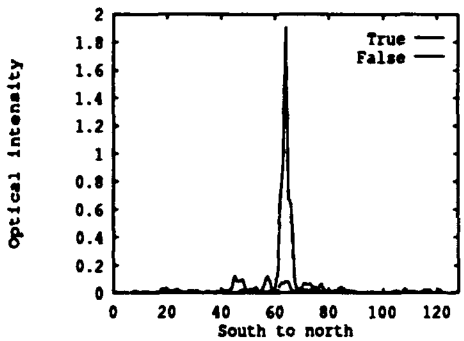

Figure 5.19 Optical intensity in the correlation plane for 15 degrees true and false image obtained for the BPOF generated by the hillclimber.

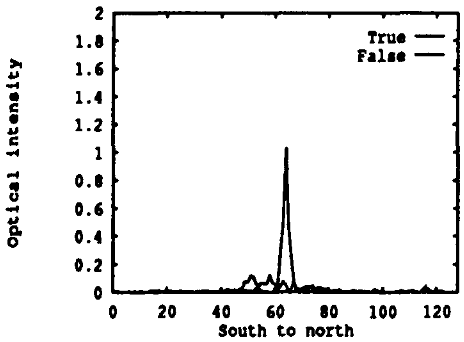

Figure 5.20 Optical intensity in the correlation plane for 20 degrees true and false image obtained for the BPOF generated by the hillclimber.

is marginally better than that obtained using Kallman's method. The minimum of the maximum of $| \langle f, h_p \rangle |^2$ inside the box B (31×31 centered in the origin) obtained for five true images is found to be 2.884 and the maximum of the maximum of $| \langle f, h_p \rangle |^2$ outside B for five true images and all through R (128×128) for five false images is found to be 0.101. In comparing this filter with that obtained using Kallman's method, this filter has a better SNR value with smaller minimum optical intensity for true images inside B and simultaneously smaller maximum optical intensity for true images outside B and for false images all through R. The SNR values obtained for M60 images at 0, 5, 10, 15, and 20 degrees from the front of the tank are 29.005, 43.942, 29.175, 29.932, and 29.691. The SNR values obtained for the M113 images at 0, 5, 10, 15, and 20 degrees from the front of the tank are 1.015, 1.009, 1.104, 0.866, and 1.012 only. Figure 5.21 shows the comparison of tne performance of the hillclimber and genetic algorithms used here. It is clear that the simulation results with GAs is still improving at the end of 150,000 function evaluations. Even though this filter is not as good as that generated by the hillclimber, GAs are still creating better filters. Because of the resource and time limitations, GA simulations are terminated at 300 generations. Figures 5.22 to 5.26 show the optical intensity at the correlation plane for all five true and false images for the best filter found by genetic algorithms.

The filter obtained via the GA is now used to find the SNR value for images of both tanks that are not used in the design. Images at two degrees from the front of the tank are not used in the design of the above BPOF. Both images are used to find the SNR value. For the M60 tank, the SNR value is found to be 6.033 and for the M113 tank, the SNR value is found to be 0.957. The optical intensities of both these images are plotted from south to north direction in figure 5.27. The figure shows that even though these images are not used in the design of the BPOF, the obtained BPOF is able to very well discriminate images from M60 and M113 tanks. A few other images from 0 degrees to 20 degrees are used to find the SNR for the above filter; in all cases, the

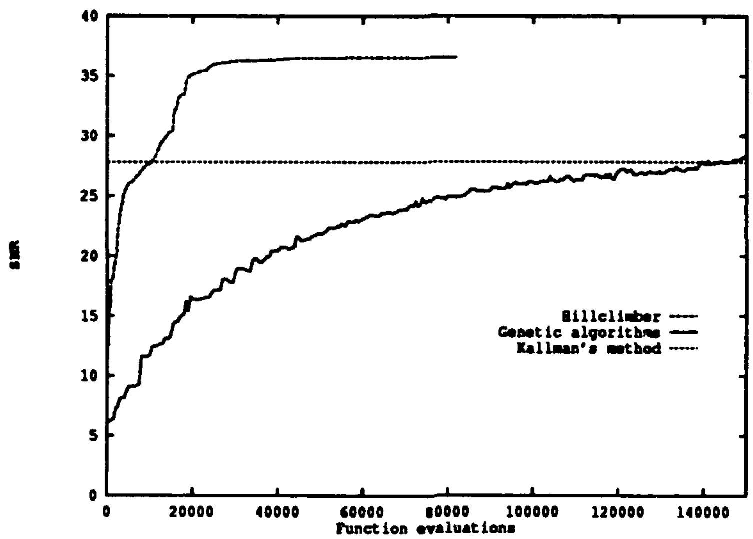

Figure 5.21 The SNR value of the best filter found by genetic algorithms and by the hillclimber versus function evaluations on distant images.

obtained BPOF is able to discriminate images from the two tanks as well.

In order to investigate the robustness of the filter for images outside the range of images used in the design, a few other images are also tried. Figure 5.28 shows the optical intensities for M60 and M113 images at 25 degrees (recall that images at only 0, 5, 10, 15, and 20 degrees from the front of the tank are used in the design) from the front of the tank. The obtained BPOF is able to discriminate the tanks as well. In the case of M60 tank, the SNR value obtained is 8.769 and in the case of M113 tank, the obtained SNR value is 1.387. However, it is observed that if images at orientation very different than those used in the filter design are used, the obtained filter is not been able to very well discriminate the tanks. Figure 5.29 shows the optical intensities of the two tanks for images at 35 degrees from the front of the tank. The SNR values for M60 and M113 tanks are 6.206 and 2.733 respectively. For images at 50 degrees from the front of the tank, the obtained BPOF has SNR values equal to 2.679 and 1.857 for M60 and M113 tanks respectively, as shown in figure 5.30.

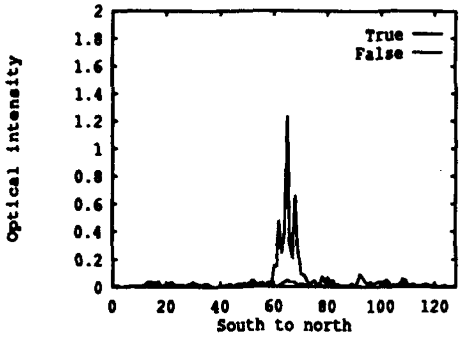

Figure 5.22 Optical intensity in the correlation plane for 0 degrees true and false images.

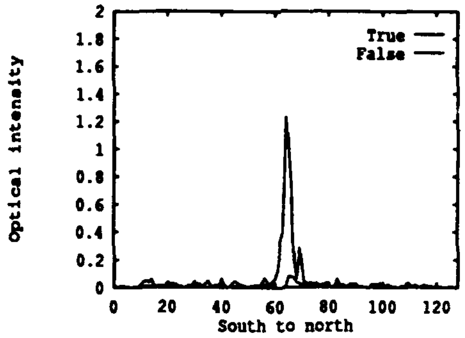

Figure 5.23 Optical intensity in the correlation plane for 5 degrees true and false images.

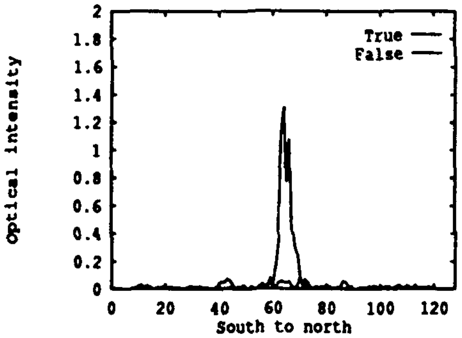

Figure 5.24 Optical intensity in the correlation plane for 10 degrees true and false images.

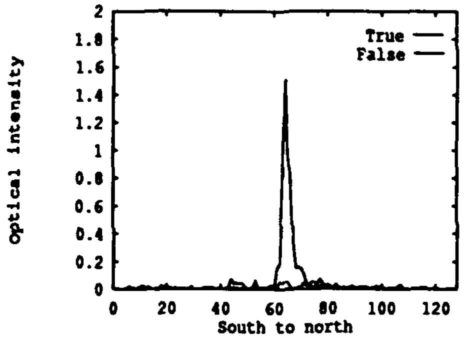

Figure 5.25 Optical intensity in the correlation plane for 15 degrees true and false images.

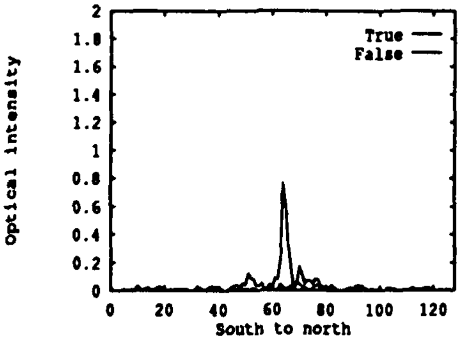

Figure 5.26 Optical intensity in the correlation plane for 20 degrees true and false images.

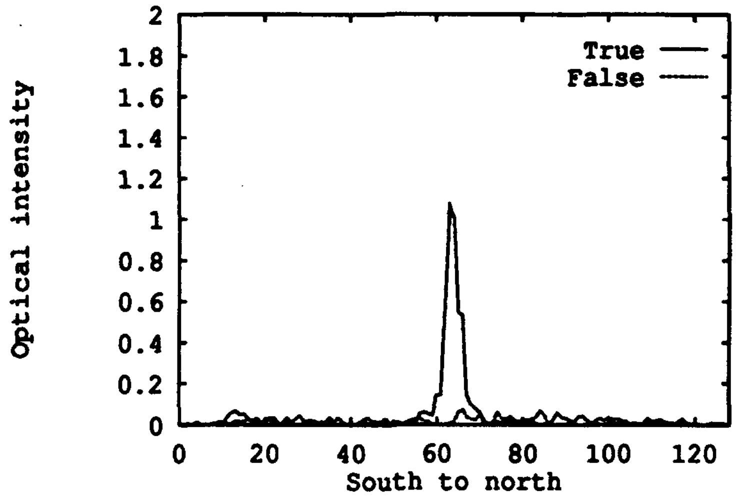

Figure 5.27 Optical intensity in the correlation plane for 2 degrees true and false images. Even though these images are not used in the design, the filter found using GAs is able to discriminate an M60 and an M113 tank.

The BPOFs found using GAs and the hillclimber are used to find the SNR value for images at other orientations. Figures 5.31 and 5.32 show that the obtained filters are not able to discriminate the tanks very well for images that are very different than those images used in the design. But for the images that are covered by the images used in the design, the obtained filters have been able to find a high SNR value. It is also interesting to note that the filter obtained using distant images has a better range of discriminating power than that obtained using consecutive filters (compare figures 5.15 and 5.31). For images used in the design, the latter has better SNR than the former, but for images that are not used in the design, the latter has worse SNR than the former. In the distant image case, the high SNR for images used for design has been sacrificed for a better spread of good SNR values over other images.

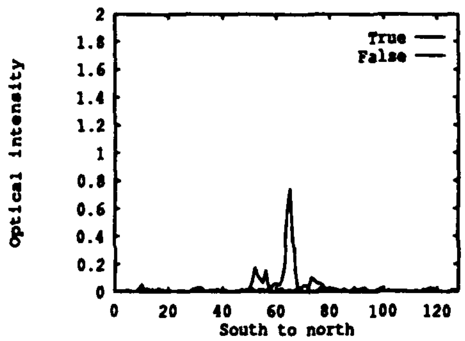

Figure 5.28 Optical intensity in the correlation plane for 25 degrees true and false images. Even though these images are not used in the design, the filter found using GAs is able to discriminate an M60 and an M113 tank.

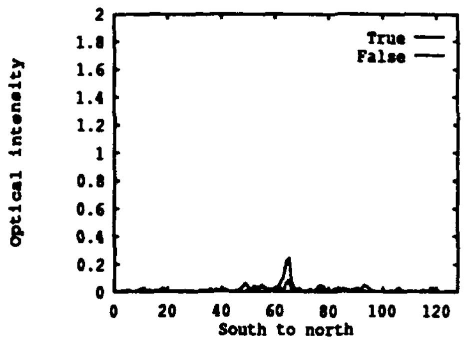

Figure 5.29 Optical intensity in the correlation plane for 35 degrees true and false images. These images are not used in the design.

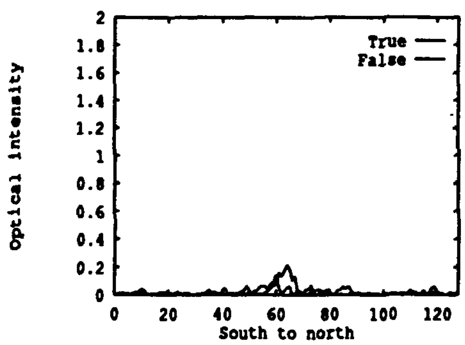

Figure 5.30 Optical intensity in the correlation plane for 50 degrees true and false images. These images are not used in the design and are too different from those used in the design to be recognized and discriminated.

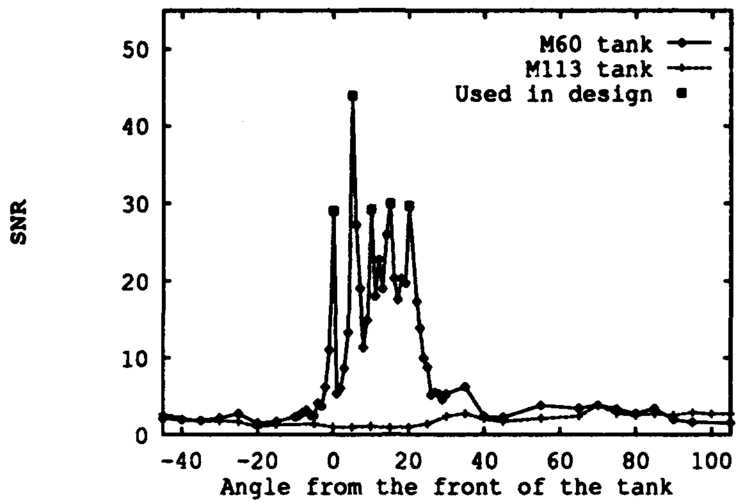

Figure 5.31 The SNR value calculated for images at various orientations using the filter obtained using genetic algorithms.

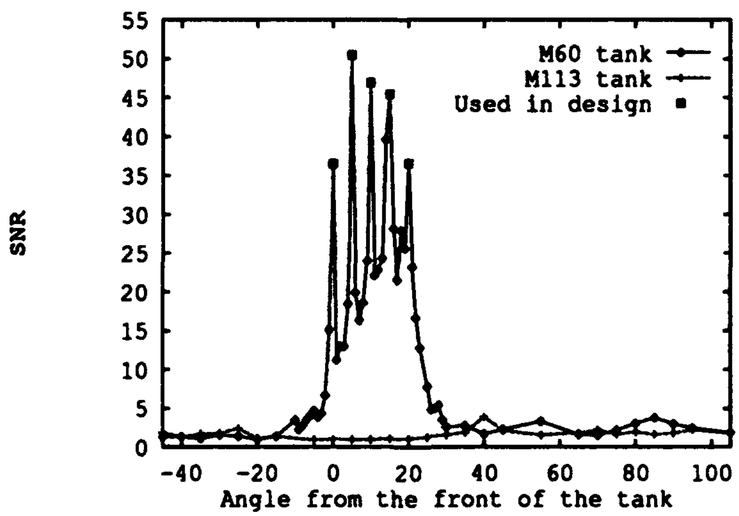

Figure 5.32 The SNR value calculated for images at various orientations using the filter obtained using the hillclimber.

5.3.3 Nine images

In the next experiment, nine true and nine false images are considered. In addition to five images at 0, 5, 10, 15, and 20 degrees from the front of the tanks, four other images at -5, -10, -15, and -20 degrees from the front of a tank are also included in the design. Only one GA simulation is performed in this case. A GA with a population size of 500, crossover probability 0.8, and five mutations per filter is used. The initial population is created from binary approximations of the matched filters of the true images. The best filter in the initial population has a SNR value equal to 1.794 and after 100 generations the best filter has a SNR value equal to 10.152.

The history of the obtained SNR as varied with function evaluations shows that the performance of GAs is improving with function evaluations. More simulations for longer generations need be done to conclude the performance of GAs on these images. Nevertheless, this experiment has shown that as more images are considered in the design, the obtained SNR for the images considered in the design could be smaller. On the other hand, as more images are considered in the design, the obtained filter has better discriminating power for a wide range of images. This trade-off between the recognition of images used in the design and discrimination of a wide range of images is important in the design of optical filters, a matter to be discussed in the next chapter.

Chapter 6

Extensions and future works

The simulations performed in this study have shown that GAs can be used to design as good (if not better) filters as by the existing method. Because of time and resource limitations, a number of investigations could not be accomplished. But this study has shown evidence to justify further and immediate investigation of different avenues of further research. In the following, a brief outline of possible future extensions is presented, followed by a brief description of each of them.

1. Include more true and false images in the design
2. Design higher-state filters
3. Use images in noisy environment
4. Design filters for amplitude-only images
5. Use other GA parameter settings
6. Use parallel GAs
7. Use GAs to find optimal initial point in Kallman's method
8. Use faster FFT routine for BPOF design
9. Estimate population sizing for knowledge-augmented initialization
10. Investigate other hillclimbers for filter design

In this study, only five true and five false images are used to design filters. There is definitely a tradeoff between the number of images used in the design versus the performance (SNR) of the filter on images. If more images are included in the design, the recognizing ability of the filter will be weaker, but the filter would be able to discriminate more images. This also raises questions about how many images should be used for optimal recognition and discrimination of images and how many filters may need to be designed for each target. The number of images to be included in the design depends on the acceptable SNR value required for the images under consideration. There should be enough included that the filter is able to discriminate well among true and false targets. The amount of computation is another concern. As the number of images increases, the computation time to design filters increases.

The idea behind designing binary phase-only filters is that matched filters are difficult to manufacture. But the binary approximation is probably the extreme discretization. Higher-state (four-state or 16-state) phase-only filters may be designed using genetic algorithms. Kallman (1990) has shown that higher-state filters produce filters with higher SNR values that that of binary filters. There are two methods by which the filters can be coded using GAs. One method would be to code the filters with higher-state alphabets. For example, in a *four-state* filter, each allele can take one of four numbers (0, 1, 2, or 3) meaning the possible phase-changes of 0, $\pi/2$, π, and $3\pi/2$. Another method would be to have a binary coding with k bits for each choice. In the above example, two bits may be reserved for the above four choices. Even though the latter coding increases the string length, more schemata may be processed with binary coding. For higher-state filters, hillclimbers may require more iterations to converge. It remains to be seen how GAs would perform on higher-state filter design, and how they would compete with hillclimbers and existing methods.

Usually the real-world images obtained from a radar or other tracking devices are embedded in some kind of noise. Random noise can be added to these images to simulate the noise effects on images and attempts made to design binary phase-only filters using GAs and using the hillclimber used here. Oftentimes, images are used by embedding them in a nonzero background. Usually a background with intensity equal to the average of the target intensities is used. This study has shown how the designed filters perform on images that are not used in the design. These filters may be tried on noisy or embedded images.

In this study, only phase-only images have been used because Kallman and Goldstein (1991) have shown that phase-only images produce better filters. Amplitude-only images or images with phase and amplitude information may be used to design filters using GAs and the hillclimber described in this study.

In this study, a number of GA parameters have been studied. There are a number of other parameters that are not varied. For example, the type of crossover operator, the probability of exchanging blocks of bits between two parents, the type of selection operator, and the type of mutation operator are a few that were kept fixed. In a two-dimensional string, the two-dimensional crossover operator used here seems reasonable from the schema survival issue; however, other crossover operators may certainly be possible. In all our simulations, only one (on average) block of bits out of nine created by the two-dimensional crossover operator is exchanged. More exchanges in the crossover operation may also be tried. In this study, only the tournament selection operator is used, since the tournament selection does not have the *scaling* problem and has better convergence characteristics unlike proportionate selection methods (Goldberg & Deb, 1991). Other selection methods may also be tried. In this study, only one-bit mutation is tried. Other types of mutation operator like changing a number of bits simultaneously may be used. Some problem related information may be used to design these operators.

Since the function evaluations take most of the computation time, a parallel GA will be very suitable for this application. The tournament selection and the two-dimensional crossover operator used here require only two strings. The mutation operator require only one string. Thus, the GA used here can be easily parallelized. With sufficient processors, the SNR calculations may be performed in parallel and thus filters can be designed in a short time.

Instead of designing a filter with phases at each of 16,384 locations as variables, GAs may be used to design a filter as in Kallman's method. Once the matched phase-only filter is found, a suitable binary approximation of the matched filter may produce an optimal binary phase-only filter. This approximation involves only one variable—a point on the unit circle used as the boundary for choosing a phase change of π or not. GAs can be used to find the optimal point on the unit circle. The filter obtained by this method may then be compared with Kallman's filters and filters obtained by methods of this study.

All the above extensions would be much easier and quicker to accomplish, if a faster Fourier transform method is used. All simulations in this study have been performed on a IBM Risc6000 system with a FFT routine (written in C) that takes about 0.3 sec for Fourier transforming a 128×128 complex array. Some of the GA and hillclimber runs performed here took about a week to complete the simulation. A faster FFT code will enable a faster investigation of the above studies in a shorter time.

Goldberg, Deb, & Clark (1992) developed an estimate for sizing an initial random population in order to detect a certain signal-to-noise in a problem. If a knowledge-augmented initial population is used, this sizing may be more than sufficient. To derive a generalized sizing equation for knowledge-augmented initial population for any problem may be difficult, but an attempt may be tried for this filter design problem. This will require knowledge about the underlying building blocks—their size and their interaction among each other—in the problem. It may be easier to use a small size problem to investigate these aspects.

It is found in this study that the simple bit-by-bit hillclimber has performed as well as the GA (sometimes better) and better than the existing method. A sophisticated hillclimber (Ackley, 1987) may create even better filters. Instead of changing one bit at a time, a number of bits may be changed at a time. Moreover, other hillclimbers—steepest ascent, next ascent, and others—may also be tried.

Chapter 7

Conclusions

In this study, binary phase-only optical filters have been designed using genetic algorithms (GAs). GAs are stochastic search methods based on natural principles. Binary phase-only filters are designed for detecting M60 tanks and rejecting M113 tanks, images of which were supplied by the Wright Laboratory Armament Directorate at Eglin Air Force Base, Florida. GA filters with tournament selection, a two-dimensional crossover operator, and a simple mutation have been found as good as the filters obtained using an existing method. A hillclimbing technique is also used to design these filters.

First, GAs have been used to design filters for recognition of a single image. A parametric study has been performed to find better GA parameter setting for this problem. GAs have found a filter that is 14% better than the matched phase-only filter. This parameter setting is then used to design filters for multiple image recognition and discrimination problems. In one case, GAs found a filter that is marginally better and the hillclimber found a filter that is 30% better than that obtained by an existing deterministic method.

The success of GAs and an hillclimbing technique used in this study opens a number of doors for further research. These preliminary results suggest that GAs can be successfully used to design better optical filters. The experiments have also shown that GAs may require more function evaluations to achieve better filters. But this is not a limitation in this problem. The filters are usually required to be designed before hand. GAs bid very well to be a promising technique for optical filter design. More research need be performed to investigate these claims. A number of future research topics in that direction are also suggested.

References

Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing.* Boston, MA: Kluwer Academic Publishers.

Antonisse, J. (1989). A new interpretation of schema notation that overturns the binary encoding constraints. *Proceedings of the Third International Conference on Genetic Algorithms*, 86–91.

Bartelt, H., & Horner, J. (1985). Improving binary phase correlation filters using iterative techniques. *Applied Optics, 16*(18), 2894–2897.

Booker, L. B. (in press). Recombination distributions for genetic algorithms. To appear in the Second Workshop Proceedings on Foundations of Genetic Algorithms.

Calloway, D. L. (1991). Using genetic algorithm to design binary phase-only filters for pattern recognition. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 422–429.

Eshelman, L. J., & Schaffer, J. D. (in press). Real-coded genetic algorithms and interval-schemata. To appear in The Second Workshop Proceedings on *Foundations of Genetic Algorithms.*

Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization, and machine learning.* New York: Addison-Wesley.

Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems, 6*, 333-362.

Goldstein, D. (October, 1992). Personal communication.

Grefenstette, J. J., & Fitzpatrick, J. M. (1985). Genetic search with approximate function evaluations. *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, 160–168.

Hester, C. F., & Casasent, D. (1980). Multivariant technique for multiclass pattern recognition. *Applied Optics, 19*(11), 1758–1761.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Horner, J. L., & Bartelt, H. (1985). Two-bit correlation. *Applied Optics, 24*(18), 2889–2893.

Horner, J. L., & Gianino, P. D. (1984a). Phase-only matched filtering. *Applied Optics, 23*(6), 812–816.

Horner, J. L., & Gianino, P. D. (1984b). Additional properties of the phase-only correlation filter. *Optical Engineering, 23*(6), 695–697.

Horner, J. L., & Gianino, P. D. (1985). Applying the phase-only filter concept to the synthetic discriminant function correlation filter. *Applied Optics, 24*(6), 851–855.

Horner, J. L., & Leger, J. R. (1985). Pattern recognition with binary phase-only filters. *Applied Optics, 24*(5), 609–611.

Kallman, R. R. (1986a). Construction of low noise optical correlation filters. *Applied Optics, 25*(7), 1032–1033.

Kallman, R. R. (1986b). Optimal low noise phase-only and binary phase-only optical correlation filters for threshold detectors. *Applied Optics, 25*(23), 4216–4217.

Kallman, R. R. (1987). Direct construction of phase-only filters. *Applied Optics, 26*(24), 5200–5201.

Kallman, R. R. (1988a). Digital and optical shape representation and pattern recognition. *Proceedings of SPIE–The International Society for Optical Engineering, 938*, 40–47.

Kallman, R. R. (1988b). Two variants of the optical correlation process. *Proceedings of SPIE–The International Society for Optical Engineering, 928*, 40–47.

Kallman, R. R. (1990). The design of phase-only filters for optical correlations. AFATL-TR-90-63, Air Force Armament Laboratory, Eglin Air Force Base.

Kallman, R. R., & Goldstein, D. H. (1991). Invariant phase-only filters for phase-encoded inputs. *SPIE–Optical Information Processing Systems and Architectures III, 1564*, 330–347.

Kargupta, H., Deb, K., & Goldberg, D. E. (1992). Ordering Genetic Algorithms and Deception. *Parallel Problem Solving from Nature*, North-Holland, 47–56.

Kim, M. S., & Guest. C. C. (1989). Simulated annealing algorithm for binary phase only filters in pattern recognition. *Applied Optics, 29*(8), 1203–1208.

Kim, M. S., & Guest. C. C. (1990). Experiments on annealed binary phase only filters fabricated with electron beam lithography. *Applied Optics, 29*(23), 3380–3386.

Mahlab, U., & Shamir, J. (1991a). *Comparison of iterative optimization algorithms for filter generation in optical correlators.* Unpublished manuscript.

Mahlab, U., & Shamir, J. (1991b). Genetic algorithm for optical pattern recognition. *Optics Letters, 16*(9), 648–650.

Mahlab, U., & Shamir, J. (1992). Iterative optimization algorithms for filter generation in optical correlators: A comparison. *Applied Optics, 31*(8), 1117–1125.

Spears, W. M., & De Jong, K. A. (1991). An analysis of multi-point crossover. *Foundations of Genetic Algorithms*, 301–315.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, 2–9.

Radcliffe, N. J. (in press). Genetic set recombination. To appear in The Second Workshop Proceedings on Foundations of Genetic Algorithms.

Vose, M. D. (in press). Generalizing the notion of schema in genetic algorithms. *Artificial Intelligence.*

Wright, A. H. (1991). Genetic algorithms for real parameter optimization. *Foundations of Genetic algorithms*, 205–218.

DISTRIBUTION LIST
(WL-TR-93-7017)

DTIC/DDAC
Cameron Station
Alexandria VA 22304-6145
1

WL/MNOI (Scientific and Tech. Info. Facility)
203 W. Eglin Blvd., Ste. 300
Eglin AFB FL 32542-6843
1

WL/MNGA
101 W. Eglin Blvd.
Attn: Dr. Dennis Goldstein
Eglin AFB FL 32542-6810
4