Testing Halftoning Methods by Images Generated by Genetic Algorithms

Timo Mantere and **Jarmo T. Alander**
University of Vaasa,
Department of Information Technology and Production Economics,
P.O. Box 700, FIN-65101 Vaasa, Finland
firstname.lastname@uwasa.fi

Abstract

This work introduces automatic test image generation by genetic algorithms for testing different halftoning methods. In general the proposed method has potential in software test data generation. This study was done in order to link together two different projects: the first one concentrates on software test data generation by genetic algorithms and the second one studied digital halftoning optimization for an ink jet marking machine also by genetic algorithm. The object software halftones images with different image filters.The goal was to reveal, if genetic algorithm is able to generate images that are difficult for the object software to halftone, in other words to find if some prominent characteristics of the original image disappear or ghost features appear due to the halftoning process. The In this work test images was created by two different ways, either using an image bitmap as a genetic algorithm chromosome, or by creating objects, such as lines, letters, rectangles, circles etc. into the image according the parameters encoded to a chromosome. results showed that genetic algorithm is indeed able to find images that are considerably changed when halftoned, and thus reveal potential problematic image features that the halftoning method do not reproduce satisfactorily.

Keywords: digital halftoning, dithering, genetic algorithms, image comparison, image filtering, software testing, test image generation.

1 Introduction

There does not seem to be much research in the field of test image evaluation. How to determine a good test image. What are the essential characteristics of a good test image? How to determine that a particular image is good for testing some specific image processing algorithm? More often than not researchers rely on commonly used and very limited test image sets. We encountered this problem, when we wanted to test the image-processing system we implemented for an ink jet marking machine[1, 2]. In our other study[3, 4, 5] we used genetic algorithms (GA) for software testing purposes. In this work we try to combine the knowledge of these two previous studies in order to use GA for generating test images for halftoning software.

This work is a continuation to that given in ref. 6. The preliminary results in that study showed that backround colour of test images was the most susceptible for distortions. This study concentrates on finding out how dominant the backround colour is and represents new implementation that has no longer just one parameter for determining the background color for each image, but several parameters that defines background segments and their colors.

1.1 Genetic algorithms

Genetic algorithms[6] are optimization methods that mimic evolution in nature[7]. They are simplified computational models of evolutionary biology. A GA form a kind of electronic population, the members of which fight for survival, adapting as well as possible to the environment, which is actually an optimization problem. GAs use genetic operations, such as selection, crossover, and mutation in order to generate solutions that meet the given optimization constrains ever better and better. Surviving and crossbreeding possibilities depend on how well individuals fulfill the target function. The set of the best solutions is usually kept in an array called population. GAs do not require the optimized function to be continuous or derivable, or even be a mathematical formula, and that is perhaps the most important factor why they are gaining more and more popularity in practical technical optimization. Today GA methods form a broad spectrum of heuristic optimization methods under intense study.

Genetic algorithm were previously adapted to the dithering problem[9, 10]. In the previous studies the human eye modulation transform function[11] is considered the best method for finding optimum halftone patterns while optimization speed in practice favors more simple methods.

For further references of GAs in image processing see bibliography 12 or book 13. Image generation with GA is used at least in ref. 14. Image generation for algorithm validation is represented in ref. 15. GAs has previously been adapted to automatic software test data generation in several studies, see refs. 3, 4, 5 and references therein.

The genetic algorithm (GA) in this study was written in Java. One of the advantages of Java is its easy to use image handling procedures. However, the execution speed of Java may not be the best possible.

1.2 Dithering

Digital halftoning[16], or dithering, is a method used to convert continuous tone images into images with a limited number of tones, usually only two: black and white. The main problem is to do the halftoning so that the bi-level output image does not contain artifacts, such as alias, moiré, lines or clusters, caused by dot placement[17]. The average density of the halftoned dot pattern should interpolate as precisely the original image pixel values as possible.

Dithering methods include static methods, where each pixel is compared to a threshold value that is obtained f.e. from a threshold matrix, generated randomly or is a static median value. Depending on matrix this method can create both frequency or amplitude modulated halftones. There are also error diffusion methods, such as Floyd-Steinberg and Jarvis-Judge-Ninke coefficients. In these methods the rounding error of the current pixel is spread into those neighboring pixels, the bi-level value of which is not yet determined.

This study concentrates only on frequency modulated halftoning methods. The three halftoning methods used here were Floyd-Steinberg (`FS`) and Jarvis-Judge-Ninke (`JJN`) error diffusions and thresholding (`TH`) with 16×16 ordered threshold matrix[16].

1.3 Comparing the images

To compare a dithered image with the original one is obviously a challenging problem. One cannot simply use pixel by pixel comparison, since dithered images usually have only two tones. The minimum difference by that measure would be achieved if every gray tone were rounded to the nearest tone (black or white), which unfortunately usually results in poor images.that usually results poor looking images in practice. Better image comparison methods have been developed[16, 18]. One alternative is to sum the pixel values from the corresponding areas ($n \times n$ window) over the images to see if the average gray tones have been preserved. With this method we can compare the images directly.

Also a set of methods called inverse halftoning[16] has been developed. From these the perhaps most common is low pass filtering, in which images are first low pass filtered and then the resulting images are compared pixel by pixel. The problem with lowpass filtering is that the high frequencies will disappear and the images get a somewhat blurred overall appearance. However, this method is easy to implement and it enables pixel by pixel comparison.

In a way the blurring by low pass filtering also resembles human eye perception: when we look the image from a distance the small details disappear and the visual observation of larger objects is averaged out from the small details.Other methods[15] to compare images have been developed, their developers suggest that these are better than simple lowpass filtering. However we have not tested these.

Another comparison method we tried is a line by line comparison of the difference between the current and the previous pixel. This method is introduced in ref. 19. For example if we take a chess board and the mirror image of it, and compare the difference between consecutive pixels in each image the result is that the two images are almost identical. The result is totally the opposite when comparing the images pixel wise.

If the images are not compared properly the received divergence value between images may as well depend on a comparison used as the actual difference between the images or the dithering methods used.

Several fitness functions i.e. image comparison methods were tested. In this study we used the average density at the corresponding image areas (`SW`), pixel by pixel comparison using low pass filtered images (`LP`), tone difference between consecutive pixels (`LS`). The fourth method used is a hybrid (`HYB`) of the three above mentioned methods. The hope was that it would take advantages of all three other but not their shortcomings. The hybrid was generated by first running those methods (index i) individually five times and then calculating the fitness gain from the best value F_{i0} from the first population to the last F_{iN}. The three methods were then weighted W_i so that the gain/best result proportion of all three methods was equal (1).

$$W_i = \frac{F_{i0}}{F_{iN} - F_{i0}} \times \frac{\sum_i F_{iN}}{F_{iN}} \qquad (1)$$

2 The proposed method

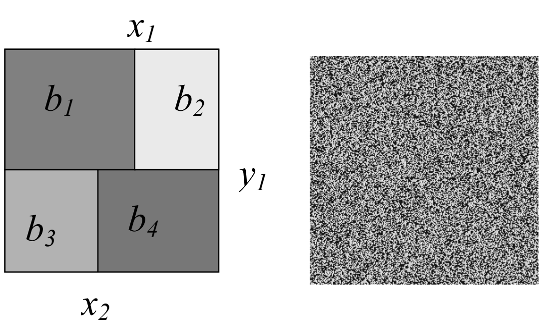
The GA runs as an independent program and optimizes parameter vectors which are used by an image generator to create images, which are further images according to that parameter array. The parameter array, one GA chromosome, is sent to image producer that generates image according them. The created image is then sent to the object software, that halftones it and returns the resulting image. The pixelgrapper reads pixels from both the test image and its halftoned transformation image and transmits 8 bit pixel arrays of both images to the fitness function evaluator. The difference between these images is used as the fitness function. GA generates new parameter vectors by using crossover and mutation, favoring those parent chromosomes that previously had gotten a high fitness value.

Test images in this study are created by optimizing parameters, such as place, size and color of elementary graphical objects, like lines, rectangles, circles and ASCII characters, together with the background tiles and colours all encoded as a GA chromosome.

2.1 Implementation

In reference 6 the coding used generated five lines, one rectangle, one circle and two characters into a test image. That coding was inflexible and results monotonic images. It required the chromosome of length 50 bytes (1 for background color, five per each line, rectangle and ellipse, seven for each character). Population size was 50, elitism 50%, total 550 evaluations (initial population + 20 generations) were done, uniform crossovers[20] was used and mutation rate was 2%.

This time the implementation used integer coded GA, where the chromosome consisted of total 79 parameters. From those the first 7 parameters were for backround, three of them (x_1, x_2, and y_1 showed in figure 1a) breaks backround into four segments and the other parameters (b_1, b_2, b_3, and b_4) determines the tone of each backround segment. This way one parameter does not dominate optimization. However, the backround might still become monotone if one sector takes whole space or the tone parameters b_x are equal.



a) b)

Figure 1:
a) Background segments b) Chaotic data to be added

The next 70 parameters where divided into 10 groups of 7 parameters, each 7 parameter long group defines one elementary image object in the following way:

1. Image object (line, rectangle, oval, ASCII character), for characters also the font style.
2. Object color.
3. Object starting point *x* coordinate.
4. Object starting point *y* coordinate.
5. Object length in *x* coordinate direction or character font size.
6. Object length in *y* coordinate direction or character font type.
7. Not used or the character value (only printable ASCII characters were used).

All objects are opaque and may cover earlier created objects, backround is created first and then the other objects on it.

The generated image as such was still quite monotonous. Normal image usually has more variation between neighbouring pixels. Our test image was further diversified by adding chaotic data (see fig. 1b) with Verhulst [21] logistic equation (2).

$$x_i = a \times x_{i-1} \times (1 - x_{i-1}) \qquad (2)$$

The chaotic data was used rather than random noise in order to control diversity and to keep the added noise repeatable. The last two parameters of the chromosone is a 16-bit value *a* for Verhulst function that was scaled to be decimal number in range [2, 4].

The optimization process usually favoured chaos parameters that generated striped patterns rather than patterns that resembles random noise (fig. 1b).

The size of the generated image was 256×256 pixels, so that the values of most parameters would fit into 8 bits. Population size was 50, elitism 40%, total 3050 evaluations (initial population + 100 generations) were done, uniform crossovers was used, and mutation rate was 1%.

The transparent implementation by using 98 parameter long chromosome was also tested, however transparent objects did not seem to bring any added value into results, so transparent implementation is not discussed further in this paper.

3 Experimental results

The results were generated by running five test runs with each one dithering method together with one comparison method, so there were altogether 12 dithering/comparison method pairs and 60 GA test runs. Each dithering/comparison method pair was also tested twice by testing as many randomly generated test images as in one GA optimization.

Five separate runs with each dithering/comparison method pair were performed in order to see if the result images have similarities after each run with same objective. The primary goal was not to find best possible result after several test runs.

The very first observation in ref. 6 was that the separate test runs with the given dithering method and image comparison method tend to produce images with nearly the same dominating background tone *b*. This time we try to find out if backround tone really explains the whole result.

The results obtained with GA optimization was compared with the results obtained three other ways. First with the randomly generated test images. Secondly testing all possible monotone images with tones between 0 and 255. In the third comparison run we tested images from the commonly used, "standard", test image set {Lena, Bird, Boat, Goldhill, Mandrill, Peppers}.

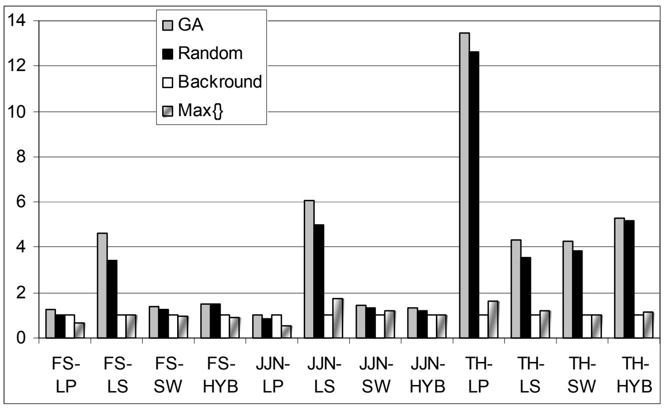


Figure 2: Comparison of the best values with different test image sets and dithering/comparison method pairs.

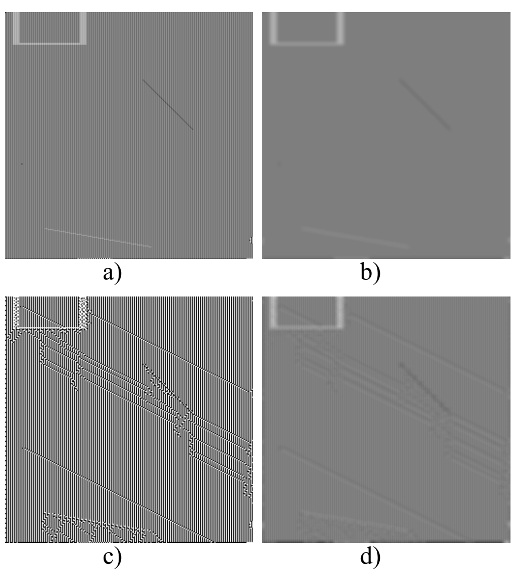


Figure 3:
a) best solution for `TH-LP`. b) low pass filtered a.
c) dithered a. d) low pass filtered c.In

Figure 4 shows a comparison of the best values obtained by using test images generated by GA (`GA`), random method (`Random`), testing all possible monotone images (`Backround`) or test image set (`Max{}`). Figure collects results from all dithering methods used in this study combined with all comparison methods, altogether 12 halftoning/comparison method pairs.

Figure 2 the best value found with each test set is divided by the best value obtained with monotone images (backround), so the best value obtained with backround test set is exactly 1 in each pair.

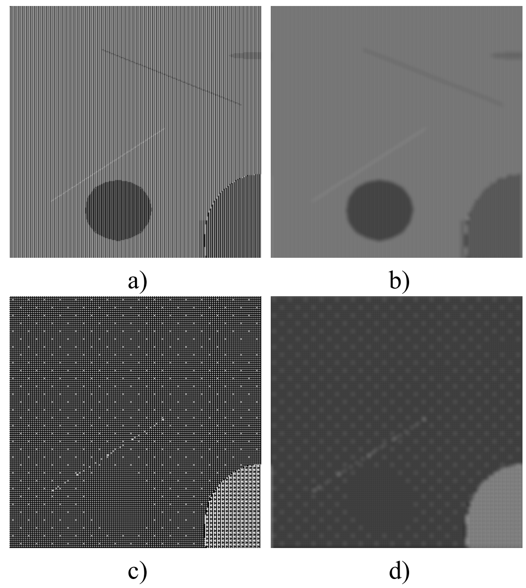


Figure 4:
a) best solution for `JJN-LS`. b) low pass filtered a.
c) dithered a. d) low pass filtered c.

From figure 4 we can see that for each of the 12 dithering/comparison method pair the GA has reached the best fitness value for each item of the four image sets.

The random method has usally generated nearly as good solution as GA. Six times GA and random method has generated images that have got considerably higher fitness value than monotonic images or images from standard test set. So the results also confirms that backround tone does not explain the whole difference score. The other objects in the image are also important for explaining the fitness result.

The biggest difference between images generated by GA and the standard test images was obtained when using threshold matrix as dithering method and comparing the dithered and the original images after low pass filtering (`TH-LP`). The large difference in this particular case seems to be due that GA is able to find weaknesses from the ordered dither matrix and generates image patterns that results considerable difference between compared images. Furthermore max{} in the image 2b refers to the value that best test image from test image set {Lena, Bird, Boat, Goldhill, Mandrill, Peppers} got in the same comparison. With all pairs GA was able to found synthetic image that got higher difference than any of the images in the test set.

With ordered threshold matrix GA was able to generate the largest differences. This seems to be due that GA detects the weak points of threshold matrix, i.e. generates dark pixels to the places where threshold values are low and light pixels to where threshold value is high. This leads to the halftone result that dramatically differs from original image, and this difference is visually very obtainable also, see fig. X.

With comparison method LA (selitys myöhemmin) GA was able to generate test images that causes large differences with all halftoning methods.

Figure 3 shows an example of how the large difference in `TH-LP` is composed. GA has found such chaos parameters that results in vertical stripes. These stripes happen to be in such position that when dithered by the ordered threshold matrix the backround becomes considerably darker.

Interestingly also the circle in the lowright corner has come much lighter. The dark circle has thus totally disappeared into background. The left hand side images are the original and the dithered, while the right hand side images are the low pass filtered from them.

Figure 4 shows an example of the typical shadow images that dithering generates. Fig. 4 is from `JJN-LS` pair that generates the second largest fitness difference between GA optimized images and the standard test images.

Figures 5a-c shows an example how the gray level histograms develop during the optimization run. The gray tones that generate higher difference between the original and the dithered image are heavily increasing. In the original population tones are fairly randomly distributed (fig. 5a). In the last generation tones are clustered around dominating tones 8 and 246 (fig. 5d).

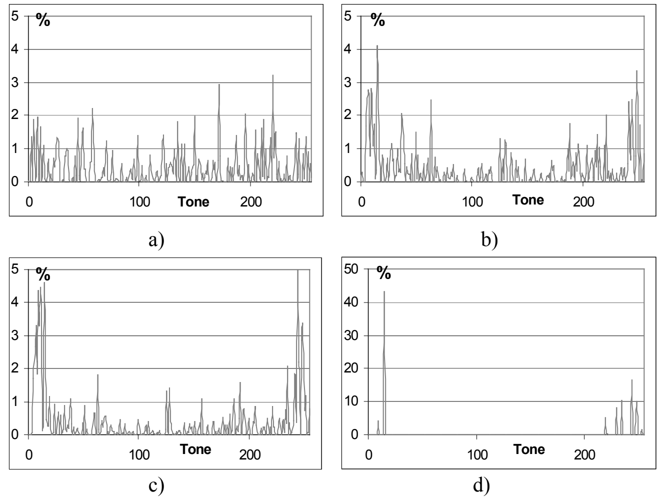


Figure 5: Gray level histogram development during optimization process
a) initial population b) 10th generation
c) final population d) the best solution found

4 Conclusions and discussion

The results seems to confirm that GA is capable of generating test images for testing different halftoning methods. Either some features of the original image disappear or some artifacts appear. The changes were perceived either by comparing the original and the dithered image, or comparing low pass filtered versions of the images. The preliminary results of this work showed that the background tone is by far the most significant factor when testing the dithering methods. However, backround tone is clearly not the only factor, this study confirms that objects in the backround usually generate larger difference than a solid one tone image.

4.1 Future

Image comparison could be enhanced by applying some feature extraction method, like MRDF[22]. Also the possibilities of applying fuzzy logic to image comparison is under study. GA coding can be improved. Integer coded GA may not be the most suitable for this problem. It is planned that future version will more freely create desired objects. More massive test runs may eliminate the bias of background tone dictation. The significance of other objects and their position in the image may be identified if we use static background tones and let other features settle.

Statistical analysis of the generated image parameters should be done in order to fully determine possible correlating parameters. So far the observations have been made by observing the images and analyzing only a couple of variables. However, so far the work is been mostly experimental, the goal has been to solve what this kind of optimization approach results in software testing, and how the method could be further improved.

After a satisfying fitness function has been found, the obvious application of the above testing method is automatic dithering method design. One GA generates halftone filters while the other GA tries to create the hardest test image for each filter. The best filter being the one where the hardest test image is closest to the original after dithering. In general this kind of GA based approach could be used in the design and testing of demanding software.

Bibliography

1. Jarmo T. Alander, Timo Mantere, and Tero Pyylampi, Threshold matrix generation for digital halftoning by genetic algorithm optimization. In David P. Casasent, editor, *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, volume SPIE-3522*, Boston, MA, 1.-6. November 1998. SPIE, pp. 204-212, 1998.

2. Jarmo T. Alander, Timo Mantere, and Tero Pyylampi, Digital halftoning optimization via genetic algorithms for ink jet machine. In B. H. V. Topping, editor, *Developments in Computational Mechanics with High Performance Computing*, CIVIL-COMP Press, Edinburg, UK, pp. 211-216, 1999.

3. Jarmo T. Alander, Timo Mantere, Ghodrat Moghadampour and Jukka Matila, Searching protection relay response time extremes using genetic algorithm – software quality by optimization. *Electric Power Systems Research 46*, pp. 229-233, 1998.

4. Jarmo T. Alander, and Timo Mantere. Automatic software testing by genetic algorithm optimization, a case study. In Conor Ryan and Jim Buckley (eds.) *SCASE'99 - Soft Computing Applied to Software Engineering*, 11.-14.4.1999, Limerick, Ireland, pp. 1-10, 1999.

5. Jarmo T. Alander, and Timo Mantere, Genetic algorithms in software testing - experiments with temporal target functions. In *MENDEL2000 6th International Conference on Soft Computing*, June 7-9, 2000, Brno, Check Republic, Brno University of Technology, PC-DIR, Brno, pp. 9-14, 2000.

6. Timo Mantere, and Jarmo Alander. Automatic test image generation by genetic algorithms for testing halftoning methods. In David P. Casasent, editor, *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision, volume SPIE-4197*, Boston, MA, 5.-8. November 2000, SPIE, Bellingham, Washington, pp. 297-308, 2000.

   Khoral Research Inc., *Digital Image Processing (DIP) with Khoros Pro 2001*, Version 3.0.1.2, March 2000. Khoral Research Inc., Albuquerque, New Mexico. Available via www: < http://www.khoral.com/contrib/contrib/dip2001/index.html >

7. John Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor, MI, Reissued by The MIT Press, 1992.

8. Charles Darwin, *The Origin of Species: By Means of Natural Selection or The Preservation of Favoured Races in the Struggle for Life*. Oxford University Press, London, A reprint of the 6th edition, 1968.

9. N. Kobayashi, and H. Saito, Halftoning Technique Using Genetic Algorithms. In *Systems and Computers in Japan **27***, pp. 89-97, Sept. 1996.

10. J. Newbern, and V. M. Bove, Jr., Generation of blue noise arrays using genetic algorithm. In *Human Vision and Electronic Imaging II*, B.E. Rogowitz and T.N. Pappas, eds., vol SPIE-3016, pp. 401-450, SPIE, Bellingham, San Jose, CA, 10.-13. Feb. 1997.

11. J. Sullivan, L. Ray and R. Miller (1991). Design of Minimum Visual Modulation Halftone Patterns. In IEEE Transactions on Systems, Man, and Cybernetics, ***Vol. 21, No 1***, January/February 1991.

12. Jarmo T. Alander, *An Indexed Bibliography of Genetic Algorithms in Optics and Image Processing*. Department of Information Technology and Production Economics. University of Vaasa. Report Series No. 94-1-OPTICS, 2000, Available via ftp:< ftp://garbo.uwasa.fi/cs/report94-1/gaOPTICSbib.ps.Z>

13. Sankar K. Pal, Arhish Ghosh, and Malay K. Kundu, *Soft Computing for Image Processing*. Physica-Verlag, Heidelberg, New York, 1999.

14. Karl Sims, Artificial Evolution for Computer Graphics. *Computer Graphics (Siggraph '91 Proceedings)*, July 1991, pp.319-328.

15. Mario Miyojim, and Heng-Da Cheng, Synthesized images for pattern recognition. In *Pattern Recognition*, Vol. 28, No. 4, pp. 595-610, 1995.

16. Henry R. Kang, *Digital Color Halftoning*. SPIE Optical Engineering Press, Bellingham, Washington, & IEEE Press, New York, 1999.

17. Peter G. J. Barten, *Contrast Sensitivity of the Human Eye and Its Effects on Image Quality*. SPIE Optical Engineering Press, Bellingham, Washington, USA, 1999.

18. Fredrik Nilsson, Objective quality measures for halftoned images. *Optics, Image, Science and Vision*, Volume 16, Number 9, September 1999, pp. 2151-2162.

19. Patrick Eklund, and Fredrik Georgsson, Unraveling the Thrill of Metric Image Spaces. In G. Bertrand, M. Couprie, and L. Perroton (editors*), Discrete Geometry for Computer Imagery*, LNCS 1586, Springer, 1999

    R.C. Gonzales and R. E. Woods, *Digital Image Processing*. Addison Wesley, 1993.

20. Gilbert Syswerda, Uniform crossover in genetic algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*. George Mason University, June 4-7, 1989. Morgan Kaufmann Publishers, Inc. San Mateo, California.

21. Paul S. Addison, *Fractals and chaos: An illustrated course*. Bristol, Philadelphia, Institute of Physics Publishing, 1997.

    John R. Koza, *Genetic Programming*. Cambridge, MA; The MIT Press, 1992.

22. A. Talukder, and D. Casasent, General methodology for simultaneous representation and discrimination of multiple object classes. *Optical Engineering, Special Issue on Advanced Recognition Techniques 37* (3), pp. 904-913, March 1998.