MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28-30, 2000, The University of Tokyo, Japan

13—33

Optimization of Image Processing by Genetic and Evolutionary Computation: How to Realize Still Better Performance

Hisashi Shimodaira
Faculty of Information and Communication, Bunkyo University
1100 Namegaya, Chigasaki-City, Kanagawa 253-8550, Japan (E-mail: shimo-hi@hi-ho.ne.jp)

Abstract

In this paper, we examine the results of major previous attempts to apply genetic and evolutionary computation (GEC) to image processing. In many problems, the accuracy (quality) of solutions obtained by GEC-based methods is better than that obtained by other methods such as conventional methods, neural networks and simulated annealing. However, the computation time required is satisfactory in some problems, whereas it is unsatisfactory in other problems. We consider the current problems of GEC-based methods and present the following measures to achieve still better performance: (1) utilizing competent GECs, (2) incorporating other search algorithms such as local hill climbing algorithms, (3) hybridizing with conventional image processing algorithms, (4) modeling the given problem with as smaller parameters as possible, and (5) using parallel processors to evaluate the fitness function.

1. Introduction

One of the main objectives of image processing is to analyze images provided by an imaging system and then to locate and recognize the object in the environment. Fig. 1 shows the fundamental steps in digital image processing for such purposes: preprocessing, edge detection and segmentation; representation and description of pattern shape and feature extraction; pattern matching, recognition and interpretation. In order to perform such processing automatically, the prior knowledge in the knowledge base is used. Alternatively, the prior knowledge can be acquired by iterative learning based on the existing data. In these areas of image processing, there are many problems whose optimum solutions need to be searched efficiently in complex solution spaces.

The genetic and evolutionary computation (GEC) is the generic name for genetic algorithm (GA), genetic programming (GP) that is the extension of GA, evolutionary strategy (ES) and evolutionary programming (EP). All these algorithms are stochastic searching processes that are inspired by the evolution of biological organs. Although these three paradigms have originated from different advocators and had peculiar characteristics of their own [1], at present, these paradigms have been merged to each other and there exist hardly clear boundaries to discriminate them. As for the details, readers should refer to [2], [3] and [4].

The outline of the simple GA is shown in Fig. 2. The population comprises a group of chromosomes that are candidates for the solution of the given problem. Initially, the population is generated randomly. The fitness value of the chromosome is obtained by evaluating the value of the objective function (fitness function) to be optimized. A particular group of chromosomes (parents) is selected from the population based on a prescribed rule to generate the offspring by the defined genetic operators, namely, mutation and crossover. The chromosomes in the current population are then replaced by their offspring, based on a certain replacement strategy to form the population in the next generation. Because the selection rule has a bias toward favoring chromosomes with a higher fitness value, the fitness value of each

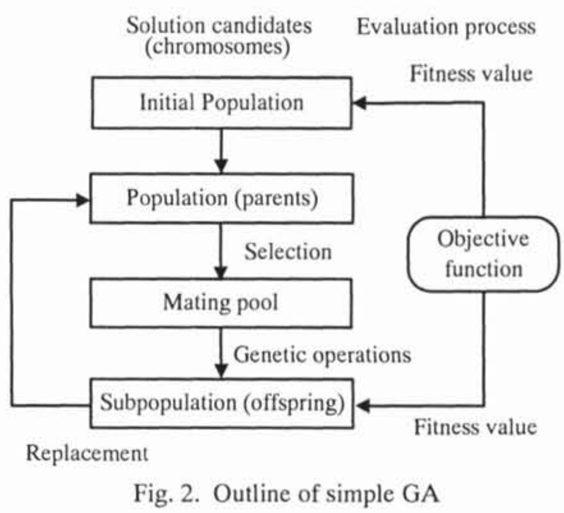

Fig. 2. Outline of simple GA

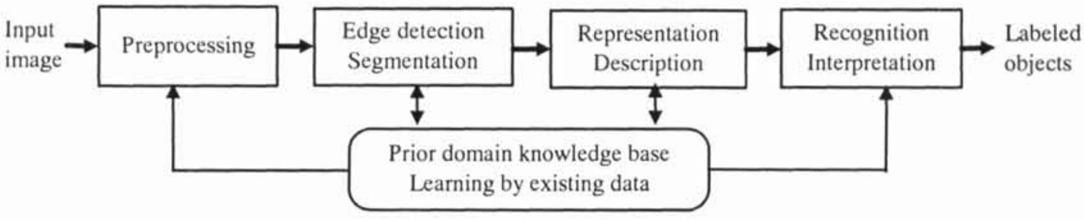

Fig.1. Fundamental steps in digital image processing

chromosome becomes higher by repeating such a cycle. This cycle is terminated when a desired criterion is reached (for example, a predefined maximum number of generations). If all goes well throughout this process of simulated evolution, the best chromosome in the final population becomes a highly evolved solution to the problem. This is the case of the canonical simple GA, whereas various competent GECs that show superior performance to the simple GA have been proposed.

The chromosome is constructed by the parameters to be optimized for the given problem. As for the representation of chromosome, in the GA paradigm, a binary-coded string has been traditionally used, whereas a real-coded string is also used nowadays. In the EA and EP paradigms, a real-coded string has been traditionally used. The objective function (fitness function) is the one that is to be maximized or minimized in the modeling of the given problem. This function takes a chromosome as input and produces a fitness value as a measure to the chromosome's performance. If the user provides the fitness value interactively in real time, based on a cognitive skill such as recognition, the algorithm can evolve a solution using this ability. Such a paradigm is called interactive GEC. The two key factors critical to the success of GEC-based methods are (1) the manner in which the possible solution to the problem is represented by the chromosome and (2) the manner in which the possible solution is evaluated in the context of the problem domain.

At present, besides GECs, we have various optimization algorithms: enumerative techniques such as branch and bound and dynamic programming; calculus-based techniques that uses the gradient-directed searching mechanism; simulated annealing (SA); neural networks (NNs), etc. Compared with other optimization techniques, GECs have the following advantages that allow us to model and implement easily the given problem as an optimization problem: (1) GECs make relatively few assumptions about the solution space; (2) GECs do not involve sophisticated objective functions and the constraints of the given problem can be easily included by describing them as penalty terms of the objective function; (3) the objective function does not need to be differentiable or continuous; (4) the interface between the GEC and the evaluation process involves only the passing of function evaluation values. On the other hand, the simple GA has a problem of poor scale-up behavior. That is, although the simple GA works well with small problems, with larger or harder problems, the solution times increase or the solution quality decreases, or both.

Recently, GECs have gained a growing popularity and a fairly great number of attempts to use GECs to solve complex problems in various application fields, including image processing, have been conducted. On the other hand, some people have raised questions about the utility of GECs as an optimization tool in the real world, due to the poor performance of the simple GA. Under such situations, in this paper, we examine the results of major previous attempts to apply GECs to image processing and outline the way of using GECs, their effectiveness and efficiency, including comparisons with other methods. Additionally, we consider the current problems of GEC-based methods and present measures to achieve still better performance.

2. Image processing using genetic and evolutionary computation

2.1. Applications of genetic and evolutionary computation in image processing

Major applications of genetic and evolutionary computation in image processing are listed in Table 1.

Table 1. Applications of genetic and evolutionary computation in image processing

Investigators	Year	Description
◆ Preprocessing		
Pall [5]	1994	Automatic selection of an image enhancement operator using GA. Chromosome: a binary-coded string that represents the 12 parameter values of a generalized enhancement function.
◆ Edge detection		
Bhandarkar [6]	1994	A GA-based optimization method to choose a minimum cost edge configuration. Chromosome: a binary-coded 2D array that represents pixels.
Harris [7]	1996	A GP technique to produce high-performance edge detectors for 1D signals and image profiles.
◆ Geometric primitive and shape detection		
Lutton [8]	1995	A GA technique to detect several geometric primitives in the same run using the sharing technique [9]. Chromosome: parameters representing the shape of primitives.
Kawanishi [10]	1995	A GA technique to detect plural kinds of shapes by interpreting differently each chromosome. Chromosome: parameters representing the shape of primitives.
Chakraborty [11]	1998	A GA in combination with the randomized Hough transform to deal with complex noisy images.
Yin [12]	1999	Detection of a circle and ellipse using a hybrid scheme that consists of a GA phase and a local search phase. Chromosome: parameters representing the shape of primitives.
Ser [13]	1999	Hough transform that makes use of GA whose fitness function is derived based on the analysis of peak formation in the 4D generalized Hough transform's parameter space.
◆ Image segmentation		
Bhanu [14]	1995	A GA-based system to learn adaptively the optimum values of the 14 control parameters of the Phoenix segmentation algorithm.
Chun [15]	1996	A GA technique to maximize the quality of segmented regions generated by a split-and-merge process. Chromosome: integers that represent region numbers obtained by the split-and-merge process.

Investigators	Year	Description
Bhandarkar [16]	1999	A hybrid algorithm that combines GAs and stochastic annealing algorithms such as simulated annealing. Chromosome: a 2D array that represents the region number of each pixel.
Cagnoni [17]	1999	A method for evolving adaptive procedures to optimize the parameters of the contour detector using GA.
◆ Shape representation		
Huang [18]	1999	A polygon approximation method based on GA. Chromosome: a binary-coded string whose bit represents a point on the objective curve.
◆ Feature selection		
Siedlecki [19]	1989	A GA technique to find the optimal subset of features from a larger set of possible features. The fitness function includes the number of dimensions to be selected and the error rate of classification.
Vafaie [20]	1992	A GA technique to find the optimal subset of features from a larger set of possible features. The fitness values are calculated by recognition rates with the AQ15 classification inducer.
Kuncheva [21]	1997	A GA technique to find the optimal subset of features from a larger set of possible features.
Yamany [22]	1997	A method that combines GA-based feature selection procedure and a neural network classifier. The fitness function includes a penalized term accounting for the cardinality of the reference set.
Kelly [23]	1991	A GA-based method for transforming data to increase the accuracy of a k-nearest neighbor algorithm. Chromosome: real number for rotation angles of the data set member and scaling factors for each attribute.
◆ Clustering and classification		
Srikanth [24]	1995	A GA-based classification method in which the class boundary is approximated by a set of fuzzy ellipsoids. Chromosome: a binary-coded string of variable-length to represent multiple ellipsoids.
Sarkar [25]	1997	An EP-based clustering algorithm that groups a given set of data into an optimum number of clusters. A structural mutation operator for adding and deleting clusters is employed to find the optimum cluster number.
Pal [26]	1998	A GA-based method of finding decision boundaries that are approximated by piecewise linear segments generated from a set of hyperplanes. Chromosome: parameter values representing hyperplanes.
Bandyopadhyay [27]	1998	An extended method of Pal's one [26] using a variable-length string GA to determine automatically the optimum number of hyperplanes.
Ishibuchi [28]	1995	A GA-based method for selecting an optimum set of fuzzy if-then rules to construct a compact fuzzy classification system with high classification power.
Iwata [29]	1996	A GA-based pattern recognition system using evolvable hardware that can change its own structure. Chromosome: variable -length binary bit string to represent the architecture of a programmable logic device.
◆ Model-based object recognition and interpretation		
Hill [30]	1992	An application of GA to the model-based image interpretation. As an example, the boundary of the left ventricular of a heart is located using a flexible template with six shape parameters.
Toet [31]	1995	A model-based matching scheme based on GAs. The size and shape of the model contour representing 2D image shapes adapt to local image evidence. Chromosome: the parameter values of the model.
Yokoo [32]	1996	A GA-based method that can detect human facial regions. The facial region is approximated by an ellipse. Plural facial regions can be detected in the same run. Chromosome: five parameter values for an ellipse.
Tsang [33]	1997	A scheme for matching and recognizing broken object boundaries. The best alignment between the model that are transformed by the affine transform and the object shapes is estimated using GA.
Ozcan [34]	1997	A method for shape recognition in which GA is applied to the partial matching. Model shapes are described in terms of features such as line segments and angles using attribute strings.
Undrill [35]	1997	An application of the GA to the model-based anatomical object recognition using a flexible template. A 3D Fourier descriptor is used to represent the model shape.
Fisher [36]	1999	A application of ES for the registration between the 3D surface model and the scene in a system for recognizing and locating rigid 3D objects.
Mignotte [37]	2000	A statistical model-based method using a hybrid GA to classify shadow shapes of man-made objects in sonar imagery. A steepest ascent technique for local search and a cooling temperature schedule is employed.
◆ Interpretation on the basis of the prior knowledge		
Ankenbrandt [38]	1990	A method for labeling complex scenes via GA. A scene is modeled by the semantic net that consists of classification categories and relationships expressed as fuzzy truth functions between categories.
Meyer [39]	1997	An application of a simple GA to the line labeling problem in the scene that is cast into optimization framework.
◆ Applications to the learning process for object recognition		
Caldwell [40]	1991	An iterative GA to evolve the composite of a criminal suspect. The selection is performed by having a witness view the generated twenty faces and rate each one according to its resemblance to a culprit.
Katz [41]	1994	A GA-based adaptive system for detecting targets in image data based on a statistical classifier. The filters to extract feature vectors are generated through the learning process using GA.

Investigators	Year	Description
Rizki [42]	1994	An adaptive pattern recognition system. that evolves cooperative sets of feature detectors and combines
Rizki [43]	1995	their response. GA and EP are employed to determine optimum morphological operators for the detectors.
Soodamani [44]	1998	A machine vision system in which a GA-based learning paradigm is incorporated in the feedback path that connects the output recognition performance to the input stage.
◆ Automatic program generation by GP		
Tackett [45]	1993	An automatic target recognition system in which GP is used to construct classifiers that process the feature vectors produced by an existing algorithm. The simulations were performed using large volumes of real data.
Andre [46]	1994	A GP-based approach to evolve a program for recognizing noisy multi-font and multi-size characters using decision rule sets. Hand-coded rule sets can be upgraded by including them into the initial population.
Johnson [47]	1995	An application of GP to the evolution of visual routines for simple tasks for machine vision.
Poli [48]	1996	A GP-based approach to develop efficient optimal image filters that can perform image enhancement, feature detection and image segmentation. The experiments were performed using two kinds of medical images.
Daida [49]	1996	A GP paradigm to discover algorithms that can extract and classify pressure-ridge features from images of arctic sea ice. The GP is used as a scaffold to support image analysts within the cycle of hypothesis-test.

2.2. Summary of the results

According to the results that are reported in the literatures mentioned above, as for the solution accuracy (quality), the GEC-based optimization methods are promising for practical use. The results of comparison with other methods reported in the literatures are summarized in Table 2. We should note the following three respects. (1) For problems that are tractable with conventional method, the solution accuracy (quality) of the GEC-based methods is significantly better than that of conventional methods. (2) For many problems that are intractable with conventional methods, excellent results are also obtained by the GEC-based methods. For example, Ser's method [13] can detect occluded objects that cannot be detected by the standard generalized Hough transform. Also with the Johnson's GP-based method [47], the evolved program shows better performance than the best algorithm written by hand. (3) In most cases, the GEC-based methods outperform non-conventional methods such as NNs and SA, whereas a competent SA outperforms the simple GA as shown in Ianni's paper [50].

Although the computation time required is not always reported in the literatures, we should note the following descriptions: (1) it is, if anything, satisfactory: Yin [10], Cagnoni [17], Huang [18], Toet [30], Tsang [32] and Mignotte [37]; and (2) it is, if anything, unsatisfactory: Ser [13], Chun [15] and Fischer [34]. When compared with other methods, (1) it is shorter than other methods: branch and bound algorithm in Siedlecki [19], gradient-based algorithm and SA in Mignotte [37], NN in Tackett [45] and local search in Whitley [52]; (2) it is almost the same as other methods: SA in Hill [35]; and (3) it is longer than other methods: local search in Bhandarker [6], generalized Hough transform in Ser [13] and SA in Ianni [50]. In general, harder and more complex problems require more computation time. Also, the computation time required is not predictable because of the stochastic nature of GECs. Especially, GP-based methods require tremendously much computation time at the training stage. Therefore, when we use GEC-based methods for practical use, we must devise techniques to reduce the computation time.

Katz [41] compared the GA-based approach with the conventional approach in the filter design and revealed that the strength of the GA-based approach is development time: the GA-based system required only a few hours to develop, whereas the conventional approach took months.

Table 2. Comparison of accuracy (quality) and evaluation (+ : GEC is superior, * : almost the same, - : GEC is inferior).

Processing	Compared method and evaluation [Reference]
Edge detection	Local search +, SA * [6] ; Canny's method + [7]; SA + [30]
Shape detection	FCQS + [10]; Generalized Hough transform + [13], Hough transform + [11], SA - [50]
Segmentation	Traditional method + [14]; Split and merge + [15]
Polygonal approximation	Traditional methods + [18]
Feature selection	Sequential search + , Branch and bound + [19]; Sequential backward selection + [20]; ID3 + [23];
Clustering	K-means algorithm +, Fuzzy K-means algorithm + [25]
Classification	NN + [24]; Bayes classifier *, k-nearest neighbors +, NN + [26]
Target recognition	Principal component method * [41]; Binary tree classifier +, NN + [45]
Object recognition	Gradient-based algorithm +, SA + [37]; Conventional method + [40]; Human + [47]; NN + [48]

3. Measures to achieve still better performance

Considering the results of the previous research, in order to achieve still better performance, we should take the following measures.

(1) It is widely known that for some complex problems, the simple GA often exhibits poor performance, especially lower performance than conventional local search algorithms, as shown in Bhandarkar's [6], Myers's [39] and Miller's [51]

papers. Various competent GAs that show better performance than the simple GA have been proposed: for example, IGA [6], messy GAs [52], Genitor [53], CHC [54], DCGA[55], etc. Therefore, in order to obtain still better solutions, we should employ one of them.

(2) It is well known that the incorporation of other search algorithm into a GA is very effective to improve the performance (convergence speed, stability and reliability) of the GA. For example, for multiple fault diagnosis problems, Miller [51] performed extensive experiments on hybrid GAs in which local improvement operators are incorporated and indicated that such hybrid GAs can find optimal solutions in most cases. Also, Bhandarkar [16] showed that the hybrid algorithm that combines GAs with stochastic annealing algorithms exhibits superior performance as compared with the simple GA. Also, Ozcan [33] incorporated a problem-specific hill climbing algorithm into the GA. Mignotte [37] incorporated a steepest ascent algorithm into the GA. Therefore, in order to obtain still better solutions, we should devise such hybrid GECs according to the given problem.

(3) Considering the tradeoff between the solution accuracy (quality) and the computation time, the hybrid of GECs and conventional image processing algorithms is a good compromise to achieve relatively better performance, as shown in Yin's [10], Bhanu's [14] and Chun's [15] papers.

(4) When the dimension of parameters to be optimized becomes larger, the optimization becomes much harder and more computation cost, especially computation time is required. Therefore, we should model the given problem as an optimization problem with as smaller parameters as possible. In this sense, it is problematic to use a 2D-array of attribute values of each pixel in the image as the chromosome. We must develop a new method for image coding to adapt the GEC structure to current technology limitations or develop a method for implementation with hardware architectures, as shown in [29].

(5) The computation time scales as $N*M$, where N is the size of the population and M is the number of generations required to obtain the solution. We can reduce M by employing competent GECs and hybrid GECs. Because the evaluation of fitness function makes up the most part of the total computation time, we can reduce the factor N by calculating the fitness values in parallel. For example, Punch [56] showed that the computation time can be reduced in inverse proportion to the number of the processors used. Therefore, in order to use GEC-based systems for practical use, we should implement them using parallel processors.

(6) Usually, GP-based methods are implemented in LISP, whereas Tackett [45] and Daida [49] showed that the version implemented in C runs about an order of magnitude faster than the LISP version. Implementing GP-based methods in C allows us to use them for practical use.

4. Conclusions

We have seen that in many problems, the accuracy (quality) of solutions obtained by GEC-based optimization methods is better than that obtained by other methods such as conventional methods, NNs and SA. However, the computation time required is satisfactory in some problems, whereas it is unsatisfactory in other problems. In general, obtaining solutions with higher accuracy (quality) requires more computation time. Therefore, we should select the method that we use from conventional methods (if available), GEC-based methods and their hybrid methods, considering the tradeoff between the solution accuracy (quality) and the computation time. We emphasize that although there is room to compare with SA and NNs, if we devise techniques to reduce the computation time, GEC-based methods have a major role to play in many problems. We feel that if we implement GEC-based methods, employing the measures mentioned above, they allow us to realize efficient and robust systems for optimizing image processing.

References

[1] T. Bäck and H.- P. Schwefel, An Overview of Evolutionary Algorithms for Parameter Optimization, *Evolutionary Computation*, 1(1), 1993, pp. 1-23.

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

[3] J. R. Koza, *Genetic Programming*, MIT Press, 1992.

[4] D. B. Fogel, *Evolutionary Computation*, IEEE Press, 2000.

[5] S. K. Pal, D. Bhandari and M. K. Kundu, Genetic Algorithms for Optimal Image Enhancement, *Pattern Recognition Letters*, 15, 1994, pp. 261-271.

[6] S. M. Bhandarkar, Y. Zhang and W. D. Potter, An Edge Detection Technique Using Genetic Algorithm-Based Optimization, *Pattern Recognition*, 27(9), 1994, pp. 1159-1180.

[7] C. Harris and B. Buxton, Evolving Edge Detectors with Genetic Programming, in *Proc. of the First Annual Conference on Genetic Programming*, 1996, pp. 309-314.

[8] E. Lutton and P. Martinez, A Genetic Algorithm with Sharing for the Detection of 2D Geometric Primitives in Images, in *Artificial Evolution*, Ed. J.-M. Alliot, Springer, 1995, pp. 287-303.

[9] D. E. Goldberg and J. Richardson, Genetic Algorithms with Sharing for Multimodal Function Optimization, in *Proc. of the Second International Conference on Genetic Algorithms*, 1987, pp. 41-49.

[10] H. Kawanishi and M. Hagiwara, A Shape Detection Method Using Improved Genetic Algorithms, in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, 1995, pp. 235-240.

[11] S. Chakraborty and K. Deb, Analytic Curve Detection from a Noisy Binary Edge Map Using Genetic Algorithm, in *Parallel Problem Solving from Nature - PPSN V*, Springer, 1998, pp. 129-138.

[12] P.-Y. Yin, A New Circle/Ellipse Detector Using Genetic Algorithms, *Pattern Recognition Letters*, 20, 1999, pp. 731-740.

[13] P. K. Ser, C. S. T. Choy and W. C. Siu, Genetic Algorithm for the Extraction of Nonanalitic Objects from Multiple Dimensional Parameter Space, *Computer Vision and Image Understanding*, 73(1), 1999, pp. 1-13.

[14] B. Bhanu, S. Lee and J. Ming, Adaptive Image Segmentation Using a Genetic Algorithm, *IEEE Trans. on Systems, Man, and Cybernetics*, 25(12), 1995, pp. 1543-1567.

[15] D. N. Chun and H. S. Yang, Robust Image Segmentation Using Genetic Algorithm with a Fuzzy Measure, *Pattern Recognition*, 29(7), 1996, pp. 1195-1211.

[16] S. M. Bhandarkar and H. Zhang, Image Segmentation Using Evolutionary Computation, *IEEE Trans. on Evolutionary Computation*, 3(1), 1999, pp. 1-21.

[17] S. Cagnoni, A. B. Dobrzeniecki and R. Poli, Genetic Algorithm-Based Image Segmentation of 3D Medical Images, *Image and Vision Computing*, 17, 1999, pp. 881-895.

[18] S.-C. Huang and Y.-N. Sun, Polygonal Approximation Using Genetic Algorithms, *Pattern Recognition*, 32, 1999, pp.1409-1420.

[19] W. Siedlecki and J. Sklansky, A Note on Genetic Algorithms for Large-Scale Feature Selection, *Pattern Recognition Letters*, 10, 1989, pp. 335-347.

[20] H. Vafaie and K. De Jong, Genetic Algorithms as a Tool for Feature Selection in Machine Learning, in *Proc. IEEE 4th International Conference on Tools with Artificial Intelligence*, 1992, pp. 200-203.

[21] L. I. Kuncheva, Fitness Functions in Editing k-NN Reference Set by Genetic Algorithms, *Pattern Recognition*, 30(6), 1997, pp.1041-1049.

[22] S. M. Yamany and K. J. Khiani, Application of Neural Networks and Genetic Algorithms in the Classification of Endothelical Cells , *Pattern Recognition Letters*, 18, 1997, pp. 1205-1210.

[23] J. D. Kelley, Jr. and L. Davis, Hybridizing the Genetic Algorithm and the K Nearest Neighbors Classification Algorithm, in *Proc. of the fourth International Conference on Genetic Algorithms*, 1991, pp. 377-383.

[24] R. Srikanth, R. George and N. Warsi, A Variable-Length Genetic Algorithm for Clustering and Classification, *Pattern Recognition Letters*, 16, 1995, pp. 789-800.

[25] M. Sarkar, B. Yegnanarayana and D. Khemani, A Clustering Algorithm Using an Evolutionary Programming-Based Approach, *Pattern Recognition Letters*, 18, 1997, pp. 975-986.

[26] S. K. Pal, S. Bandyopadhyay and C. A. Murthy, Genetic Algorithms for Generation of Class Boundaries, *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 28(6), 1998, pp. 816-828.

[27] S. Bandyopadhyay, C. A. Murthy and S. K. Pal, Pattern Classification Using Genetic Algorithms: Determination of H, *Pattern Recognition Letters*, 19, 1998, pp. 1171-1181.

[28] H. Ishibuchi and K. Nozaki, Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems*, 3(3), 1995, pp. 260-270.

[29] M. Iwata, I. Kajitani and H. Yamada, A Pattern Recognition System Using Evolvable Hardware, in *Parallel Problem Solving from Nature - PPSN IV*, Springer, 1996, pp. 761-770.

[30] A. Hill and C. J. Taylor, Model-Based Image Interpretation Using Genetic Algorithms, *Image and Vision Computing*, 10(5), 1992, pp. 295-300.

[31] A. Toet and W. P. Hajema, Genetic Contour Matching, *Pattern Recognition Letters*, 16, 1995, pp. 849-856.

[32] Y. Yokoo and M. Hagiwara, Human Faces Detection Method Using Genetic Algorithm, in *Proc. of IEEE International Conference on Evolutionary Computation*, 1996, pp. 113-118.

[33] P. W. M. Tsang, A Genetic Algorithm for Affine Invariant Recognition of Object Shapes from Broken Boundaries, *Pattern Recognition Letters*, 18, 1997, pp. 631-639.

[34] E. Ozcan and C. K. Mohan, Partial Shape Matching Using Genetic Algorithms, *Pattern Recognition Letters*, 18, 1997, pp. 987-992.

[35] P. E. Undrill and K. Delibasis, An Application of Genetic Algorithms to Geometric Model-Guided Interpretation of Brain Anatomy, *Pattern Recognition*, 30(2), 1997, pp. 217-227.

[36] D. Fischer and P. Kohlhepp, An Evolutionary Algorithm for the Registration of 3-d Surface Representation, *Pattern Recognition*, 32, 1999, pp. 53-69.

[37] M. Mignotte, C. Collet and P. Perez, Hybrid Genetic Optimization and Statistical Model-Based Approach for the Classification of Shadow Shapes in Sonar Imagery, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(2), 2000, pp. 129-141.

[38] C. A. Ankenbrandt and B. P. Buckles, Scene Recognition Using Genetic Algorithms with Semantic Nets, *Pattern Recognition Letters*, 11, 1990, pp. 285-293.

[39] R. Myers and E. R. Hancock, Genetic Algorithm Parameter Sets for Line Labeling, *Pattern Recognition Letters*, 18, 1997, pp. 1363-1371.

[40] C. Caldwell and V. S. Johnston, Tracking a Criminal Suspect through "Face-Space" with a Genetic Algorithm, in *Proc. of the 4th International Conference on Genetic Algorithms*, 1991, pp. 416-421.

[41] A. J. Katz and P. R. Thrift, Generating Image Filters for Target Recognition by Genetic Learning, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(9), pp. 1994, pp. 906-910.

[42] M. M. Rizki, and M. A. Zmuda and L. A. Tamburino, E-MORPH: A Two-Phased Learning System for Evolving Morphological Classification Systems, in *Proc. of the 3rd Annual Conference on Evolutionary Programming*, 1994, pp. 60-67.

[43] M. M. Rizki, L. A. Tamburino and M. A. Zmuda, Evolution of Morphological Recognition Systems, in *Evolutionary Programming IV*, MIT Press, 1995, pp. 95-106.

[44] R. Soodamani and Z. Q. Liu, Learning Fuzzy Modeling through Genetic Algorithm for Object Recognition, in *Proc. of IEEE International Conference on Evolutionary Computation*, 1998, pp. 656-659.

[45] W. A. Tackett, Genetic Programming for Feature Discovery and Image Discrimination, in *Proc. of the 5th International Conference on Genetic Algorithms*, 1993, pp. 303-309.

[46] D. Andre, Learning and Updating Rules for an OCR System Using Genetic Programming, in *Proc. of the First IEEE Conference on Evolutionary Computation*, 1994, pp. 462-467.

[47] M. P. Johnson and P. Maes, Evolving Visual Routines, in *Artificial Life IV*, Ed. R. A. Brooks, MIT Press, 1995, pp. 198-209.

[48] R. Poli, Genetic Programming for Feature Detection and Image Segmentation, in *Evolutionary Computing*, T. C. Forgarty Ed., Springer, 1996, pp. 110-125.

[49] J. M. Daida, J. D. Hommes and T. F. Bersano-Begey, Algorithm Discovery Using the Genetic Programming Paradigm: Extracting Low-Contrast Curvilinear Features from SAR Images of Arctic Ice, in *Advances in Genetic Programming*, Volume 2, Ed. P. J. Angeline, MIT Press, 1996, pp. 417-442.

[50] M. D. Ianni and R. Diekmann, Simulated Annealing and Genetic Algorithms for Shape Detection, *Control and Cybernetics*, 25(1), 1996, pp. 159-175.

[51] J. A. Miller and W. D. Potter, An Evaluation of Local Improvement Operators for Genetic Algorithms, *IEEE Trans. on Systems, Man, and Cybernetics*, 23(5), 1993, pp. 1340-1351.

[52] D. Whitley, J. R. Beveridge and C. Guerra-Salcedo, Messy Genetic Algorithms for Subset Feature Selection, in *Proc. of the 7th International Conference on Genetic Algorithms*, 1997, pp. 568-575.

[53] D. Whitley, The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproduction Trials is Best, in *Proc. of the Third International Conference on Genetic Algorithms*, 1989, pp. 116-121.

[54] L.J., Eshelman, The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, in *Foundation of Genetic Algorithms*, 1991, pp. 265-283.

[55] H. Shimodaira, A Diversity-Control-Oriented Genetic Algorithm (DCGA): Development and Experimental Results, in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-99) Volume 1*, 1999, pp. 603-611.

[56] W. F. Punch, E. G. Goodmann and M. Pei, Further Research on Feature Selection and Classification Using Genetic Algorithms, in *Proc. of the 5th International Conference on Genetic Algorithms*, 1993, pp. 557-564.