A Novel Genetic Programming Algorithm for Designing Morphological Image Analysis Method[*]

Jun Wang and Ying Tan[**]

Key Laboratory of Machine Perception (Ministry of Education), Peking University
Department of Machine Intelligence, School of EECS, Peking University, P.R. China
wangjun@cis.pku.edu.cn ytan@pku.edu.cn

Abstract. In this paper, we propose an applicable genetic programming approach to solve the problems of binary image analysis and gray scale image enhancement. Given a section of original image and the corresponding goal image, the proposed algorithm evolves for generations and produces a mathematic morphological operation sequence, and the result performed by which is close to the goal. When the operation sequence is applied to the whole image, the objective of image analysis is achieved. In this sequence, only basic morphological operations— erosion and dilation, and logical operations are used. The well-defined chromosome structure leads brings about more complex morphological operations can be composed in a short sequence. Because of a reasonable evolution strategy, the evolution effectiveness of this algorithm is guaranteed. Tested by the binary image features analysis, this algorithm runs faster and is more accurate and intelligible than previous works. In addition, when this algorithm is applied to infrared finger vein gray scale images to enhance the region of interest, more accurate features are extracted and the accuracy of discrimination is promoted.

1 Introduction

Digital images processing on computers have been applied to many fields like pattern recognition, robotic vision, biomedical image analysis, and biometrics, etc [10][12][16]. Recent years, a variety of evolutionary methods are used to solve the problem of discovering algorithm for image processing [2][7][11][13]. Most of them concentrate on that the image analysis problems can be re-framed as filtering problems, and use genetic algorithm (GA) and genetic programming (GP) to produce a set of standard filters [7][11][12]. However, all these filter-based GA(GP) methods need complicated formulations which require a large amount of analysis and computation [11].

Mathematic morphological is a powerful non-linear tool for extracting image components, which is useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hulls [5]. Morphological operators aim at

[*] This work is supported by National Natural Science Foundation of China (NSFC), under grant number 60875080 and 60673020, and partly supported by the National High Technology Research and Development Program of China (863 Program), with grant number 2007AA01Z453.

[**] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2011, Part I, LNCS 6728, pp. 549–558, 2011.

extracting relevant structures of the image considered as a set through its sub graph representation, which is achieved by probing the image with another set of the known shapes called *structural element* (SE) [15]. Searching suitable morphological operation sequence and corresponding SEs in a big searching space is suited to the genetic programming [13]. Some researchers use an evolutionary morphological approach to analyze image. Yoda et al. explore the possibility of obtaining mathematic morphological algorithm for binary images by means of GA [18]. Harvey et al. describe a technique by GA for the optimization of multidimensional gray scale soft morphological filters [6]. They use this technique in the spatiotemporal domain for applications in automatic film restoration. Quintana et al. propose an approach of morphological binary image analysis based on GP [13][14]. Their algorithm is constructed by logic operators and the basic morphological operators — erosion and dilation — with a group of manually chosen SEs. This algorithm evolves to generate morphological operation sequence which converts a binary image into the target image contained just a particular feature of interest. Additional, they prove that it is possible to evolve good morphological methods by using GP. Ballerini et al. propose a GP method without a goal image and the morphological operations are not used for noise reduction or segmentation, but for image classification [1].

In this paper, we propose a applicable genetic programming algorithm to automatic generate methods for binary image analysis and gray scale image enhancement. Given a section of original image and the corresponding goal image, this algorithm automatically produces a mathematic morphological operation sequence, and the result by which is close to the goal. Afterwards, when the operation sequence is applied to the whole image, the objective of image analysis is achieved. This paper is organized as follows. Section 2 presents a novel genetic programming algorithm with mathematic morphological operations. The experimental results of binary images analysis and gray scale image enhancement are reported in Section 3. Finally, Section 4 gives the concluding remarks of this paper.

2 Proposed GP Algorithm

2.1 Definition of GP Algorithm

The genetic programming provides an approach to the problem of finding a computer program to solve a problem [9]. In our problem, by giving an original image and a learning target— goal image, the GP algorithm automatically produces a sequence of operations, which is applied to the original image and the obtained result is very close to the goal.

Definition of Chromosome. A chromosome in this algorithm is an individual, i.e., a method, which is composed by a sequence of genes, illustrated in Fig 1.

We use gene to express the primitive operation set. One single gene is a primitive unit of morphological operation accompanied with logical operations. The number of genes in each chromosome may be different while each gene has a fixed length. And two processing sequences are kept in one chromosome, which are connected by logical operations.

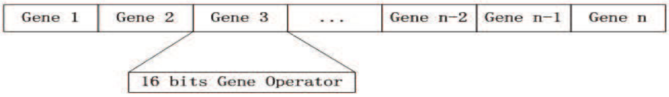

Fig. 1. Chromosome definition

2.2 Definition of Gene

The gene structure is shown in Table 1. Each gene has 16 bits and only 14 bits have real meanings since the first two bits are used for alignment. All these bits are classified into two parts: one part expresses logical operation and the control flow, and the other part expresses morphological operation.

Table 1. The meaning of each bit in a gene

	Index	Meaning
Unused	1	null
	2	null
logical and control	3	switch
	4	storage flag
	5	direct/difference flag
	6	logical operation flag
	7-8	logical operator
morphological operation	9	erosion/dilation flag
	10-11	SEs size
	12	SE class
	13-16	SEs index

Direct/Difference flag. The direct/difference flag means which one will be chosen as current output, the morphological operation result directly or the arithmetic difference between the result and the input. This operation brought forward will improve the morphological expression of this algorithm. In the morphological meanings, the arithmetic difference devotes to morphological gradient [15].

Storage flag. The storage operation was firstly introduced in [14] represents whether the result of this gene should be stored or not. When the finite automaton parses a chromosome, the input of current gene is the previous one's result and the output is the input of next one. Therefore, the intermediate result is abandoned. The storage operation means whether memorizes the intermediate result, which can be used in subsequent logical operations.

Logical operation. The logical operator flag represents which logical operation will be applied to current gene. Four logical operations are defined in this algorithm, *AND, OR, NOT* and *XOR*. Some logical operations need two operands: one operand is current morphological operation result, and the other one is the stored intermediate result mentioned above. When being applied on the binary images, the logical operations are bitwise. There are some changes when the logical operations are applied to gray scale images [15].

Switch operation. In this paper, we bring forward a new operation— "switch", which helps to keep two operation sequences in one chromosome. When the automaton parses a chromosome, there is a register named "backup", which is initiated with the source image and changed by the switch flag. If the switch flag is true, the input of current gene is exchanged with the backup. Therefore, the backup is brought foreground to be processed in this section, and the current input is stored as the backup. The switch operation accompanied with storage operation can express two operation sequences in one chromosome. Actually, many morphological operations are hardly realized in one process [15]. Because of two operation processes contained in one chromosome, this algorithm can express most of the morphological operations with basic erosion and dilation [15].

Pattern of structural elements. Erosion and dilation are basic morphological operations [5][15], which depend on the pattern of SE. The pattern space of SE is exponentially increased along with its size . For example, a SE whose size is $n \times n$ has $2^{n \times n}$ type of patterns. Therefore, we define the patterns manually. We use the concept of *regular and irregular* SEs same as [13], but with some differences: regular(irregular) SEs are defined which are all symmetrical(asymmetrical, vice versa). Three sizes of SEs are used in our algorithm, 3×3, 5×5 and 7×7. Each size has two styles, regular and irregular, and each style has 16 patterns. Patterns of 3×3 SEs are shown in Fig 2.

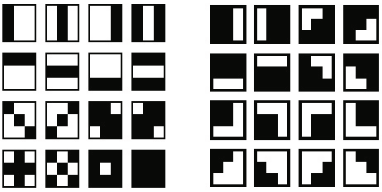

Fig. 2. Patterns of 3×3 *regular & irregular* structural elements

Fitness function. The objective fitness function $F(0 \leqslant F \leqslant 1)$ is known as similarity as the correlation coefficient between a processed image and the goal [18],

$$F = \frac{(f \cdot g)}{\sqrt{(g \cdot g)} \times \sqrt{(f \cdot f)}}$$

where f and g are two binary images of size $M \times N$ and

$$(f \cdot g) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} f(i,j) \cdot g(i,j),$$

image f is the processed result, and image g is the goal.

Since in a binary image the white pixels represent objects, this is a reasonable choice of fitness function. The optimum is $F = 1$ when all the pixels match. The worst case is $F = 0$ when none of the pixels match [18].

2.3 Evolution Strategy

We define a structural evolution strategy in this algorithm. This algorithm initiates 1024 chromosomes at the beginning which are all single genes. The chromosome will grow in evolution and its length is limited to no more than 20 genes. The selection rate is 0.3, the mutation rate is 0.3, the crossover rate is 0.4, max generations is 300.

Structural mutation. Each gene has 14 valid bits and the random variant range is $2^{14} = 16348$. Therefore, it is hard to evolve effectively if the gene are mutated randomly. We bring forward a structural mutation strategy. All these 14 bits in one gene are divided into three parts according to their function: flow control, logical operation and morphological operation. These three parts are mutated separately, which makes the mutation position well–distributed.

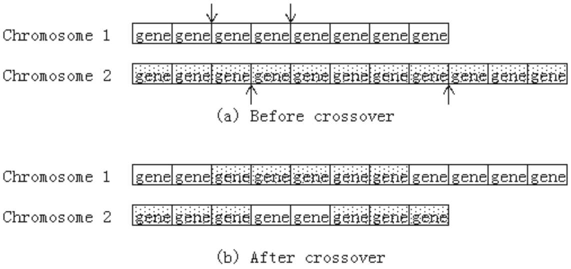
Chromosome 1
gene gene gene gene gene gene gene gene
Chromosome 2
gene gene gene gene gene gene gene gene gene gene gene
(a) Before crossover
Chromosome 1
gene gene gene gene gene gene gene gene gene gene gene
Chromosome 2
gene gene gene gene gene gene gene gene
(b) After crossover

Fig. 3. Crossover

Structural crossover. The genetic operation of crossover(sexual recombination) allows new individuals to be created [9]. We use a structural crossover in this algorithm. The basic unit of crossover is gene and crossover in the middle of a gene is not allowed. The goal of the structural crossover is to exchange a bunch of genes of two participated chromosomes. The crossover generates new individuals which may have different lengths from their parents. There are two random variants in each chromosome which participate crossover, the start position and the end position (Fig. 3). And these variants result in three different meanings:

- If the amount of one of the switched parts is zero, this means that a section of the counterpart should be inserted in it.
- If both amounts of switched parts are not zero, this means to switch sections of two chromosomes, as shown in Fig 3(b).
- If the insert position is the beginning or the end of a chromosome, this means to extend this chromosome.

3 Experimental Results

We conduct three experiments to verify our algorithm. First, we apply this algorithm to two binary image analysis experiments: artificial objects extraction and OCR music sheet analysis. These tests have been used by former researchers and we compare the effectiveness of our algorithm with other other works. Next, we apply this algorithm to gray scale image enhancement to check the effects.

3.1 Experiment on Artificial Binary Images

We use this algorithm to obtain valid objects in an artificial image composed by four features: squares, disks, rings and stars, same as Quintana et al. in [13]. Each target image has one type of the four features with randomly distributed position with size of 640 × 480. All four target images are overlaid to obtain the source image(Fig 4). The features randomly distributed in the source image may overlap, which makes their detection more difficult.

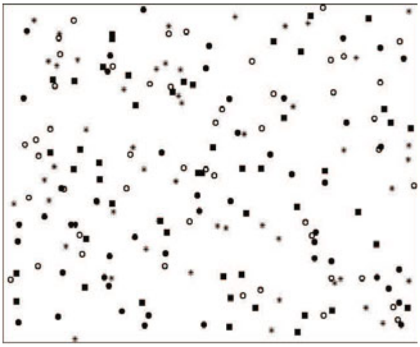

Fig. 4. Source image of artificial dataset

Training course. We select a small area from the source image which contains all four features as the source of the training set. The same areas on the target images are also selected as target images in the training set, as shown in Fig 5.

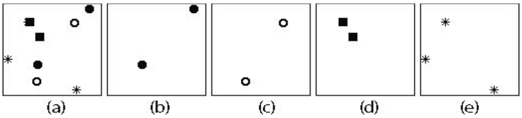

Fig. 5. Selected area of source and target images. a: source image, b c d e: target images

Testing course. The methods of four features detection are obtained in the training course by our algorithm. We apply them on the whole image, and four features are extracted separately. Executing these procedures for 10 times, we get the mean performance of our algorithm (Table. 2). The criterion is the fitness values of processed results and the target images. Except the extraction of rings, our algorithm is better than Quintana's on squares, disks and stars. Furthermore, Quintana et al. use a Linux cluster with one master node (CPU dual Intel Xeon 2 GHz, 2 GB Memory) and 22 client nodes (CPU Dual Athlon MP 1900+, 1.6 GHz, 1 GB Memory) in their experiments [13]. Our algorithm performs on a PC (CPU Intel Core Duo T9400 2.53 GHz, 4 GB Memory). Even on the most time consuming training of star, our algorithm get the result in a few minutes. This indicates our algorithm is more effective and fewer computational load.

Table 2. Performance comparison of our GP algorithm with Quintana et al.'s

Feature	Our algorithm	Quintana et al.'s
Disks	**0.9822**	0.868
Rings	0.8583	**0.906**
Squares	**0.9900**	0.870
Stars	**0.9590**	0.922

3.2 Experiment on OCR Music Sheet

We apply our GP algorithm to another type of binary image — the OCR music sheet, which was used by Yoda et al. in [18] to explore the automatic acquisition of morphological procedures by GA. We capture this music sheet from the original paper. The object of this test is to extract the four features, heads, hooks, staff lines and stems from the music sheet. Similarly, a small typical area is selected from the original image as the source image which contains all the four features. Four target images are obtained by manually calibrating, and each of which only contains one feature of four types. In the training cource of heads detection, Yoda et al. get the final fitness value of 0.963 [18], and our algorithm gets 0.9662— more accurate. The best fitness values of heads, staffs lines and stems all converge fast and stable. The best fitness value of hooks goes through a longer period, and converges to 0.9780. When we apply the methods obtained to whole music sheet, all four features are detected accurately. Experimental results also support that the efficiency and effectiveness of our algorithm are all better than Yoda et al.'s.

3.3 Experiment on Gray Scale Image

In this process, we apply this algorithm to the enhancement of gray scale image. There are more 2000 low quality finger vein images collected by an infrared CCD device, which contains 400 fingers, and each finger has 4 to 6 images. We want to use this algorithm to enhance these images and expect to extract more accurate vein features. We randomly choose four images from data set as learning samples and manually calibrate the corresponding feature images as learning goals, and all these images compose four pair of "training set". It is difficult to evaluate the effects of gray scale image enhancement. Because of the purpose of image enhancement is to get more accurate finger vein features, we use an threshold and thinning method to get the skeleton feature results of enhanced images which will be used in the evaluation. The GP algorithm runs for four times with these training set, each time with a pair of source and target, and produces four methods.

We get four operation sequences in the training course, namely morphological image analysis method(MAIM $1 \sim 4$, corresponding to four pairs of training set). Fig. 6 illustrates the four pairs of training set and the training results, images in column *a* are original images, in *b* are corresponding learning goals, in *c* are the learning results followed by threshold and thinning, and in *d* are the results by using the contrastive method which is processed by mean filtering with threshold and thinning. From the comparison, the generated methods enhance the images and more accurate features are extracted.

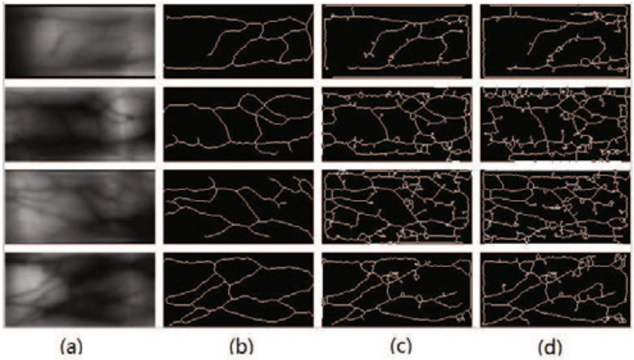

Fig. 6. Visual effect comparison for different learning pairs: (a) four original learning samples; (b) corresponding learning goals; (c) learning results of MAIM $1 \sim 4$ with threshold and thinning;(d) results obtained by mean filter with threshold and thinning

After, the obtained methods are applied to the whole data set and we examine the effects of the enhancement and verify the features obtained from enhanced images in the application of identity authentication. We test this process with combination of different number and order of MAIMes. We use the false acceptance rate (FAR) to evaluate our proposed algorithm, which is the most commonly used measure of identity authentication, the fraction of access attempts by an un–enrolled individual that are nevertheless deemed a match [17]. The classifier is Nearest Neighbor [4] and the experimental strategy is Leave–One–Out. We use the Modified Hausdorff Distance(MHD) [3][8] to measure the similarity of two images. The results are shown in Fig. 7. Bar 1 is FAR result of the contrastive method and the others are FARs of different combined MAIMes.

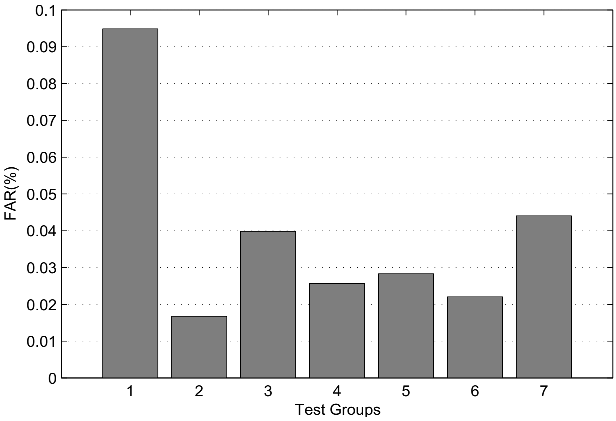

Fig. 7. Comparison of false acceptance rate for test groups and the control group. Bar 1, FAR result of control group. Bar 2, FAR result of MAIM-1,2,3,4. Bar 3, FAR result of MAIM-4,3,2,1. Bar 4, FAR result of MAIM-1,2,3. Bar 5, FAR result of MAIM-1,3,4. Bar 6, FAR result of MAIM-1,2,4. Bar 7, FAR result of MAIM-4,2,1.

Fig. 7 indicates that the enhancement methods produced by our algorithm all decrease the FARs. The most remarkable is the Bar 2 which FAR result of images are enhanced by MAIM-1,2,3,4(sequence-dependent) reduces the FAR by about 8%.

4 Conclusion

For a long time, researchers explore GA and GP approach in searching automatically produce morphological image processing methods. But the heavy computational load of evolution prevents the application of this approach. In this paper, we propose an applicable genetic programming approach to solve the problems of binary image analysis and gray scale image enhancement. It has strong ability of generalization, and shows robustness in experiments.

References

1. Ballerini, L., Franzén, L.: Genetic optimization of morphological filters with applications in breast cancer detection. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 250–259. Springer, Heidelberg (2004)
2. Daida, J.M., Hommes, J.D., Bersano-Begey, T.F., Ross, S.J., Vesecky, J.F.: Algorithm discovery using the genetic programming paradigm: extracting low-contrast curvilinear features from sar images of arctic ice, pp. 417–442 (1996)
3. Dubuisson, M.-P., Jain, A.: A modified hausdorff distance for object matching. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 1, pp. 566–568 (October 1994)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
5. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston (2001)
6. Harvey, N., Marshall, S.: The use of genetic algorithms in morphological filter design. Signal Processing: Image Communication 8(17), 55–71 (1996)
7. Hong, J.-H., Cho, S.-B., Cho, U.-K.: A novel evolutionary approach to image enhancement filter design: Method and applications. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(6), 1446–1457 (2009)
8. Huttenlocher, D., Klanderman, G., Rucklidge, W.: Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(9), 850–863 (1993)
9. Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge (1992)
10. Langdon, W.B., Poli, R., McPhee, N.F., Koza, J.R.: Genetic programming: An introduction and tutorial, with a survey of techniques and applications. In: Fulcher, J., Jain, L.C. (eds.) Computational Intelligence: A Compendium. Studies in Computational Intelligence, vol. 115, pp. 927–1028. Springer, Heidelberg (2008)
11. Munteanu, C., Rosa, A.: Gray-scale image enhancement as an automatic process driven by evolution. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34(2), 1292–1298 (2004)
12. Poli, R.: Genetic programming for image analysis. In: GECCO 1996: Proceedings of the First Annual Conference on Genetic Programming, pp. 363–368. MIT Press, Cambridge (1996)

13. Quintana, M.I., Poli, R., Claridge, E.: On two approaches to image processing algorithm design for binary images using GP. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoWorkshops 2003. LNCS, vol. 2611, pp. 422–431. Springer, Heidelberg (2003)
14. Quintana, M.I., Poli, R., Claridge, E.: Morphological algorithm design for binary images using genetic programming. Genetic Programming and Evolvable Machines 7(1), 81–102 (2006)
15. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer-Verlag New York, Inc., Secaucus (2003)
16. Tackett, W.A.: Genetic programming for feature discovery and image discrimination. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 303–311. Morgan Kaufmann Publishers Inc., San Francisco (1993)
17. Weaver, A.: Biometric authentication. Computer 39(2), 96–97 (2006)
18. Yoda, I., Yamamoto, K., Yamada, H.: Automatic acquisition of hierarchical mathematical morphology procedures by genetic algorithms. Image Vision Comput. 17(10), 749–760 (1999)