

A Genetic Algorithm Based Method for Determining Two-level Morphological Operators

Marta M. Dornelles and Nina S. T. Hirata

Department of Computer Science

Institute of Mathematics and Statistics, University of São Paulo

São Paulo, Brazil

Email: mmdornelles@gmail.com, nina@ime.usp.br

Abstract—In two-level training of binary morphological operators, several operators based on different subwindows of a larger window are designed in the first level of training, and then their outcomes are combined in a second level of training. It has been shown that such training scheme yields two-level operators with better performance than the ones designed directly on the large window. A difficult part of the two-level training scheme is, however, to set up the number of first level operators and their respective windows. To avoid the need to specify these parameters, we propose the use of a genetic algorithm based approach. Experimental results show that the proposed method can find two-level operators with better performance than those manually designed by experienced users.

Keywords—*morphological operator; image operator training; multilevel training; genetic algorithm.*

I. INTRODUCTION

Morphological operators [1], [2] are widely used in image processing and analysis. Good morphological solutions depend on adequate composition of some simple operators, and also on the choice of appropriate structuring elements. Since such task requires an expert level knowledge and a considerable trial and error effort, it has motivated research on approaches based on training techniques. Several studies have already been developed in order to design morphological operators from training data [3], [4], [5]. In its basic formulation as presented in [4], the design problem is cast as a machine learning problem in which training data is obtained from input-output pairs of images through a window W . Window W delimits the local input information to be considered to determine the output value at the same pixel location. In the context of morphological operators, these operators based on a window W are called W -operators and they are expressed as combinations of two basic operators, erosions and dilations. Window W is the parameter that defines the maximum size of the structuring elements. On one hand, large windows W are desirable because they can better discriminate local shapes. However, on the other hand, it has been observed that too large windows tend to present poorer performance due to precision issues.

To overcome this difficulty, a recent work proposed a multilevel training model [6]. Considering two-level training, the model may be described as a procedure in which several operators are designed in the first level of

training and then their outcomes are combined in the second level of training. The multilevel training model has proved to be an effective way of overcoming the difficulties for designing operators based on large windows.

To design two-level operators, not only the number of operators but also their respective windows in the first level of training must be specified. These parameters define the two-level training architecture. In [6], the number of first level operators as well as their respective windows were chosen manually, based on previous experience. A first attempt, based on information theory concepts, to automatize the choice of these parameters was proposed in [7]. However, no conclusive results concerning comparison with human expert designed operators is presented.

In this work, we propose the use of Genetic Algorithm (GA) for choosing a two-level architecture. More precisely, we would like to determine the number of first level operators and their respective windows, such that the corresponding two-level operator presents small error. Previous works that use GA based approaches for morphological operator design [8], [9], [10] are limited to a constrained class of operators (usually a subset of the W -operators, or a rather small window W).

This paper is organized as follows. In Section II, we review the two-level training procedure of W -operators. In Section III, we present the details of the main idea of this work, namely how the two-level design problem is cast as a GA based search problem. In Section IV, we present some experimental results and discuss the results. In Section V, we present our concluding remarks.

II. BINARY MORPHOLOGICAL OPERATOR TRAINING

Let $E = Z^2$. A binary image defined on E can be seen as a subset of E : if $x \in E$ is in the set, it is a foreground pixel; otherwise it is a background pixel.

Let W be a non empty subset of E containing the origin, $P(W)$ the power set of W , and $\psi: P(W) \rightarrow \{0, 1\}$. A mapping Ψ between binary images can be defined, for all $x \in E$ and $S \subseteq E$, by

$$x \in \Psi(S) \Leftrightarrow \psi(S_{-x} \cap W) = 1, \quad (1)$$

where S_{-x} denotes the translate of set S by $-x$. Equation (1) means that a given point x will be in the foreground of the transformed image $\Psi(S)$ if, and only if, the

neighborhood pattern around x in S , restricted to W , is mapped to 1 by function ψ (for convenience, we may also write $[\Psi(S)](x)=1$). Operators that can be defined as (1) are called W -operators. Designing such operators is equivalent to designing the local functions ψ .

Given observed input images S and their respective ideal output images I , we would like to find an image operator Ψ such that $\Psi(S)$ is as close as possible to I . The closeness is usually measured in terms of mean absolute error (MAE). Letting X denote a pattern observed through W on S at a given position, and y the corresponding pixel value in I , and assuming stationarity, the pair (X,y) can be seen as a random realization of a random vector (\mathbf{X},y) . Then the MAE of a W -operator Ψ , characterized by function ψ , is given as the expected value:

$$MAE(\Psi) = E[|\psi(\mathbf{X}) - y|]. \quad (2)$$

The optimal MAE operator can be characterized in terms of conditional probabilities $P(y|X)$ as follows [3]:

$$\psi_{opt}(X) = \begin{cases} 1, & \text{if } P(y=1|X) > 0.5 \\ 0, & \text{if } P(y=1|X) \leq 0.5. \end{cases} \quad (3)$$

In the basic training methodology [4], pairs of input-output images are used to estimate the joint input-output conditional probabilities in (3). A simplified scheme of the procedure is shown in Fig. 1(a), where generalization aims to determine the output value for the patterns that are not observed in the training images.

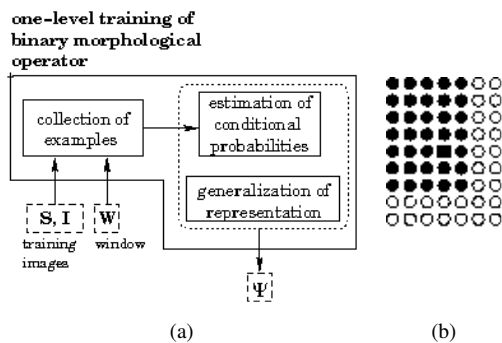


Figure 1. (a) Basic training scheme for W -operators; (b) A 7×5 window (filled circles) within a 9×7 rectangle.

All windows considered in this work are defined within a rectangular area, called window domain, centered at the origin of E . Windows will be depicted as sets of filled circles as in Fig. 1(b). Non-filled circles help to show the position of the window relative to the origin (drawn as a square).

In two-level training, operators in the first level are trained based on the training scheme shown in Fig. 1(a). Each of these operators is then applied to other training images, generating the images to be used in the second level of training. In the second level training, information from these images are combined and a second-level operator is generated. In this work, only one pixel of each image is considered in the second level of training. For example, if we consider three operators, $\Psi_1^{(1)}$, $\Psi_2^{(1)}$, and $\Psi_3^{(1)}$, in the first level, and S as an image used in the second level of training, then the training examples are of the form $([\Psi_1^{(1)}(S)](x), [\Psi_2^{(1)}(S)](x),$

$[\Psi_3^{(1)}(S)](x)$). Fig. 2 shows the scheme for two-level training of W -operators.

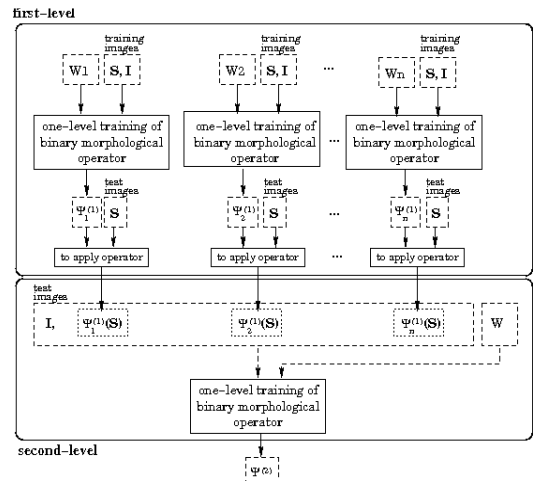


Figure 2. A scheme for two-level training of W -operators.

III. CHOOSING A TRAINING ARCHITECTURE BASED ON A GENETIC ALGORITHM

In [6], manually chosen two-level architectures with all the windows of the first-level operators defined within a larger window W are compared against one-level operators designed on W . The size and shape of W are chosen to be the ones that presented the best results in previous experiments on the corresponding datasets. It is shown that, two-level architectures that result in operators with better performance than the single-level operators designed on the large window W is found without great effort.

Nevertheless, choosing a single window W is already a task that requires some experience, and choosing the two-level architecture (i.e., how many first-level operators and their respective windows) is a more complex task. Our proposal is to use a GA based approach to determine a good training architecture.

A. Algorithm description

GA is a probabilistic search technique based on the natural evolutionary process. It is usually applied to solve optimization or search problems. Possible solutions are modeled as chromosomes and the optimality of the solution is characterized by a fitness function. An initial population of chromosomes is generated randomly or through constructive techniques, and then it is successively updated based on operators such as reproduction mechanisms, crossover and mutation [11].

The proposed method is described next.

1) *Structure of the chromosomes*: A chromosome is represented as a set of windows, each one corresponding to a window of the first-level operators. An example of chromosome with three windows is shown in Fig. 3. We assume that the size of a chromosome (i.e., the number of windows) can vary between fixed values min and max . The second level operator is not encoded in the chromosome because a fixed way of combining outputs of the first-level operators is considered in this work (as described in Section II).

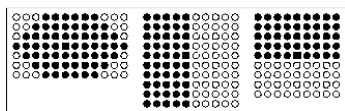


Figure 3. A chromosome with three windows.

2) *Fitness Function*: Since the aim of the design process is to find operators with small error, we use the MAE as the fitness function. Given a chromosome, its fitness is evaluated by computing the MAE of the operator encoded by it. In order to do that, previously computed results of the first level operators on a subset of training images are used to train a second level operator. The resulting two-level operator is applied on independent test images in order to estimate the MAE.

We decided to consider a pool of windows and to precompute and keep the first level operators for these windows because training of W -operators is in general a time consuming process. Windows that compose a chromosome are chosen from this pool, and thus only the second-level operator is trained in order to compute the fitness function. Since only one pixel from the output of each first-level operator is considered, second-level training is usually very fast.

3) *Population*: Both initial population size as well as chromosome size range are parameters of the algorithm. Offspring for the next generation are results of replication, crossover or mutation. The fittest chromosome of a generation is always replicated to the next generation. Population size may vary from one to another generation.

4) *Crossover*: Involves two chromosomes randomly selected according to the crossover probability (CP) which is the probability a given chromosome will undergo a crossover. A crossover point is randomly chosen in both chromosomes and the right parts of crossover points are exchanged between the pairs. If exchange duplicates a window, that window is not exchanged.

5) *Mutation*: The mutation probability (MP) is the probability a given chromosome will suffer a mutation. Two types of mutation are considered. In the first type, a randomly chosen number (between 1 and chromosome size) of windows in the selected chromosome are chosen randomly to be replaced. They will be replaced respectively by windows selected randomly from the pool of windows. However, a replacement occurs only if the selected window is not yet in the chromosome. If no replacement is made, then the second type of mutation is applied: a random number of windows (up to complete max windows in the chromosome) chosen from the pool of windows are added to the chromosome. This increases the size of the chromosome.

6) *Stopping Criterion*: Population is updated generation after generation until some stopping criterion is met. In our case, updating stops either when the maximum number of iterations is achieved or when the

fittest individual is the same for a given number of consecutive generations.

IV. EXPERIMENTAL RESULTS

In order to assess the proposed approach, we performed some computational experiments on previously tested datasets. All experiments were run on a machine with four processors of 2.40GHz each, with Debian GNU/Linux operating system.

A. Experiment description

The datasets considered in this work are described in Table I and correspond to four datasets previously cited in [6]. These datasets were chosen in order to compare our results with the results of manually chosen windows, presented in that work. We allowed the size of chromosomes to vary from $min=3$ to $max=15$.

TABLE I. DATASET DESCRIPTION

Dataset	Description	Number of training images		Number of test images
		Level 1	Level 2	
Text_s	Recognition of character "s"	6	4	5
Map	Segmentation of regions with a given texture	2	1	5
Circ	Segmentation of circular objects from diagrams	5	3	6
Box	Segmentation of rectangular objects from diagrams	5	3	6

Three types of experiments were performed:

1) *Experiment I (EI)*: the set of manual windows used in [6] were included in the pool of windows, and a chromosome formed with those windows were included in the initial population. By doing so, we guarantee that a result at least as good as the manual solution will be found.

2) *Experiment II (EII)*: the set of manual windows used in [6] were included in the pool of windows, but no individual corresponding to that set were included in the initial population. This experiment was designed to assess whether the algorithm would be able to find a solution equivalent to the manual solution.

3) *Experiment III (EIII)*: a larger pool of 315 windows were considered. All manually chosen windows for the four datasets in [6] as well as other windows within the 12×7 domain, defined without any special criterion, were included in the pool of windows. No individual corresponding to the manual choice in [6] was included in the initial population.

In EI and EII, CP values 0.5, 0.6, 0.7, and 0.8 were used and, for each of them, MP values 0.001, 0.01 and 0.1 were used. We also tested different values for the initial population size, maximum number of generations, and maximum generations without update of the fittest individual. Altogether, forty-eight different combinations of these parameter values have been tested. The pool of windows considered had around 30 windows, and the window domain used was the large window used in [6].

In EIII, parameter CP was set to 0.8, MP to 0.01, with 20 chromosomes in the initial population, maximum number of generations to 200, and the maximum number of generations without update of the fittest individual to 50.

B. Results and discussion

The best fitness found in EI, EII and EIII by GA compared to the fitness of manually chosen windows as presented in [6] are shown in Table II. In all cases, the GA found a set of windows that resulted in an operator with smaller MAE than the ones corresponding to the manually chosen windows.

TABLE II. FITNESS (MAE) FOR MANUAL WINDOWS AND THE BEST RESULTS OF GA

Dataset	Manual windows	EI	EII	EIII
Text_s	0.005520	0.003707	0.003707	0.003990
Map	0.123947	0.109482	0.108832	0.014969
Circ	0.007562	0.006063	0.005879	0.006258
Box	0.079702	0.074540	0.074470	0.045121

As expected, both EI and EII confirm that a priori knowledge on windows, such as their ideal size, helps the GA to find a good chromosome. Furthermore, EIII shows that the GA based method can find a good architecture even when the windows are not restricted to a pre-established window domain.

Table III describes the experiments that provided the best operator for each dataset. Fig. 5 presents a piece of a test image from Map dataset and the corresponding output using manually and GA selected operators.

TABLE III. INFORMATION ABOUT BEST GA RESULTS.

	Text_s	Map	Circ	Box
Type of experiment	EI	EIII	EII	EIII
Max. generations	100	200	200	200
Max. generations without change of the best chromosome	20	50	40	50
CP	0.5	0.8	0.7	0.8
MP	0.1	0.01	0.001	0.01
Initial population size	10	10	10	10
Final population size	15	39	24	35
Generations	94	193	106	158
Runtime (minutes)	30.6	303.86	93.14	245.75
Number of windows in the GA solution	7	13	8	10
Number of windows in manual solution	5	3	6	6

V. CONCLUSION

We proposed an approach based on GA to define an architecture for the two-level training of binary morphological operators.

The proposed method has been tested on four different datasets and results show that it provides a good architecture even without any information on window domain. The results also suggest that prior information on the window domain can help the GA to find better results, in fewer numbers of iterations.

The GA runtime is strongly related to the evaluation of the fitness function. As proposed, evaluation requires the training of the second level operator followed by MAE estimation on test images. Windows in the first

level could be randomly generated rather than randomly chosen from a given pool of windows. However, fitness function evaluation would become computationally prohibitive due to the need to train first level operators. In order to avoid such time consuming processing, other ways to evaluate the fitness of a chromosome will need to be devised.

To conclude, results support GA as an effective and promising way of choosing a two-level training architecture of morphological operators. In all datasets, the results obtained by GA outperformed the corresponding manual solution provided by an experienced designer.

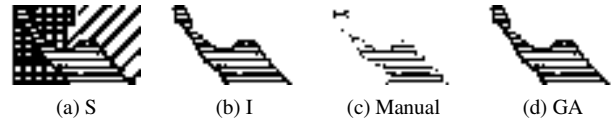


Figure 4. Example of a test image and results for dataset Map.

ACKNOWLEDGMENT

This work has been developed within the scope of a research project supported by CNPq/Brazil. N.S.T. Hirata is partly supported by CNPq/Brazil and M. M. Dornelles is supported by Universidade Estadual de Santa Cruz (UESC), Brazil.

REFERENCES

- [1] J. Serra., Image analysis and mathematical morphology, Academic Press, 1982.
- [2] P. Soille, Morphological image analysis, 2nd ed. Berlin: Springer-Verlag, 2003.
- [3] E. R. Dougherty, "Optimal mean-square N-observation digital morphological filters I. optimal binary filters," CVGIP: Image Understanding, vol. 55, no. 1, pp. 36-54, January 1992.
- [4] J. Barrera and E. R. Dougherty and N. S. Tomita, "Automatic programming of binary morphological machines by design of statistically optimal operators in the context of computational learning theory," Electronic Imaging, vol. 6, no. 1, 1997.
- [5] N. S. T. Hirata and M. M. Dornelles, "The use of high resolution images in morphological operator learning," Proceedings of the XXII Brazilian Symposium on Computer Graphics and Image Processing, pp.141-148, 2009.
- [6] N. S. T. Hirata, "Multilevel training of binary morphological operators," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 31, no. 4, pp. 707-720, April 2009.
- [7] C. S. Santos, N. S. T. Hirata and R. Hirata Jr., "An information theory framework for two-stage binary image operator design," Pattern Recognition Letters, vol. 31, no.4, pp.297-306, 2010.
- [8] R. Araujo, F. Madeiro, R. de Sousa, and F. Pessoa, "Modular morphological neural network training via adaptive genetic algorithm for designing translation invariant operators," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, 2006.
- [9] M. I. Quintana, R. Poli, and E. Claridge, "Morphological algorithm design for binary images using genetic programming," Genetic Programming and Evolvable Machines, vol. 7, no. 1, pp. 81-102, 2006.
- [10] M. Hamid, N. Harvey, and S. Marshall, "Genetic algorithm optimization of multidimensional grayscale soft morphological filters with applications in film archive restoration," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 5, pp. 406-416, May 2003.
- [11] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Professional, 1989.