PSO-Designed Operators for Image Edge Detection

Victor Ayala-Ramirez, Ezequiel Martinez-Ayala, and Raul E. Sanchez-Yanez

Universidad de Guanajuato, DICIS, Department of Electronics Engineering,
Carr. Salamanca-Valle, Km. 3.5+1.8, 36700, Salamanca, Mexico
E-mail: {ayalav,sanchezy}@salamanca.ugto.mx,ezequiel@laviria.org
(Paper received on November 28, 2010, accepted on January 28, 2011)

Abstract. We present here a method that automatically synthesize complex spatial operators from the evolution of a particle swarm and a thresholding value to perform edge detection on images. The proposed approach optimizes the values of the support of a complex spatial filter in order to emulate the response of a given edge operator on a training image. We present the results of using both a generic Canny edge operator and a Sobel edge response on a training image and the evaluation of the evolved edge operator on a set of test images. We measure the performance of edge operators using a Precision-Recall metric. First experiments show that the evolved filters are qualitatively good and that PSO can be used to emulate image processing tasks. We present both qualitative and quantitative results supporting this.

Keywords: Particle Swarm Optimization; Complex spatial filter; Edge detection;

1 Introduction

Particle Swarm Optimization (PSO) is an evolutionary computation technique originally proposed by Kennedy and Eberhardt in 1995 [3] that is useful for nonlinear function optimization in discrete and continuous spaces. PSO technique is inspired from the social behavior of schools of fishes and flocks of birds. In particular, the technique emulates how each organism moves in coordination with all their companions when they look for food or for a refuge.

There are some works in the literature where PSO has been used to perform image processing and computer vision tasks. For example, Ye *et al.* [8] have proposed to use PSO as an alternative to methods for threshold selection techniques, like the one by Otsu. Some other authors (e.g. Wei *et al.* [7], Tang *et al.*, Zhang and Liu and Zheng *et al.*) propose the utilization of PSO in image segmentation tasks. Zheng *et al.* [10] applyy a PSO-based method to the segmentation of CT and MRI images. Zhang and Liu [9] dealt with the problems arising in underwater applications. Tang *et al.* [2] address the problem of multilevel thresholding by posing it in terms of PSO.

Even edge detection has been addressed by some authors. For example, Alipoor *et al.* [1] have proposed a method to synthesize a $2D$ spatial filter as in this work. Nevertheless, they address the problem by using a real-valued mask. Another approach to edge detection has been presented by Setayesh *et al.* [6]. They propose to perform a PSO on homogeneity measures and a uniformity factor of a contour curve.

In this work, we propose to simultaneously design a complex-valued edge operator and to optimize the thresholding level in order to emulate the response of a known edge operator on a training set of images. For this purpose, we show how to emulate a Canny edge operator and a Sobel edge operator. The fitness of each particle is evaluated by using the F metric (proposed by Martin *et al.* [5]). Our experiments over a set of testing images from the Berkeley database [4] show the usefulness of the proposed method.

Rest of this work is organized as follows: in Section 2, we review the edge detection problem in terms of a PSO-based approach. Section 3 presents the application of the method to the design of two optimal edge operators. Performance evaluation of the implemented system is addressed in Section 4 as a demonstration of the validity of the proposed approach. Finally, Section 5 presents our conclusions about this work and the main aspects to be covered in future work.

2 Formulation

2.1 Edge detection

Edge detection is an essential task for image processing. This task is typically performed by convolving a $2D$ spatial operator M (defined over a given spatial support) to the input image I. The resulting image O can be expressed as:

$$O = I \otimes M \tag{1}$$

with $\otimes$ being the convolution operator. The edge detection operator M is typically characterized by a set of coeffiecients that are the spatial weights used to perform the $2D$ convolution operation. When the edge detection operation must provide a binary output, a thresholding step is needed. This process requires to set a threshold value T to decide if a pixel in the result image is considered as an edge or not.

2.2 PSO for finding an optimal edge detection operator

The objective of this work is to find the optimal parameters of an edge detection operator with respect to a training set of images $S = \{S_1, S_2, \ldots, S_m\}$. For each image S_k on the training sets S, we have a corresponding image $R_k \in R = \{R_1, R_2, \ldots, R_m\}$, that is the expected result of the edge detection process on image S_k. The elements in R can be obtained by the application of a reference operator to the images in the training set, or by any other arbitrary process, such as, the generation of the images by an expert.

In the case of binary edge detection tasks, the PSO algorithm will optimize the spatial filter coefficients of the mask M and the thresholding value T. We

$$M = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_8 & m_9 & m_4 \\ m_7 & m_6 & m_5 \end{bmatrix}$$

Fig. 1. Spatial localization of the weighting coefficients for the mask M.

consider a complex spatial mask M with complex coefficients $m_i \in C$, $i = \{1, \ldots, 9\}$. The localization of the spatial coefficients in the mask is shown in Figure 1. The thresholding value T is a real number $T \in [0, I_{max}]$ with I_{max} being the maximal intensity level hat can be present in an image.

Each particle P represents then a set of parameters for the mask coefficients and for the thresholding level. For a given particle P_j, a feasible solution of the optimization problem, we can obtain an output image O_j when a training image I is applied. O_j is computed as follows:

$$O_j = T(I \otimes M) \tag{2}$$

$$O_j(x, y) = \begin{cases} 1 & \text{if } (I \otimes M)(x, y) > T \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$O_j(x, y)$ being the pixel at spatial position (x, y) of the output image O_j.

Let O_k^P be the result of applying the edge operation process defined by the parameter set encoded by the particle P to the training image S_k. That is.

$$O_k^P = T_P(S_k \otimes M_P) \tag{4}$$

$$O_k^P(x, y) = \begin{cases} 1 & \text{if } (S_k \otimes M_P)(x, y) > T_P \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The fitness score $f(P)$ of a particle P is defined as:

$$f(P) = \sum_{k=1}^{m} F(O_k^P, R_k) \tag{6}$$

with $F(A, B)$ being a similarity metric between images A and B. We will describe below what is the metric $F(\cdot)$ used.

The optimal particle P^* will encode a parameter set consisting of an optimal mask M^* and an optimal threshold value T^*, such that:

$$P = P^* \iff f(P^*) = \max_P \sum_{k=1}^{m} F(O_k^P, R_k) \tag{7}$$

The optimal edge detection process defined by the optimal particle P^* genes can then be used to detect edeges on any input image.

2.3 Similarity Metric Between Images

Martin *et al.* [5] have proposed the F-metric to measure similarity between two edge images A and B. This methodology takes an image, let us say A, as the reference image for the comparison. In this image, the F metric only considers the edge pixels present on it. For the compared image B, we count three quantities:

TP the number of true positive edge pixels. That is, the number of edge pixels that appear both in A and B.
FP the number of false positive edge pixels. That is, the number of edge pixels that appear in B but that are not present in A.
FN the number of false negative edge pixels. That is, the number of edge pixels that appear in A but that are not present in B.

These values are used to compute two figures of merit P (Precision) and R (Recall), accordingly to:

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

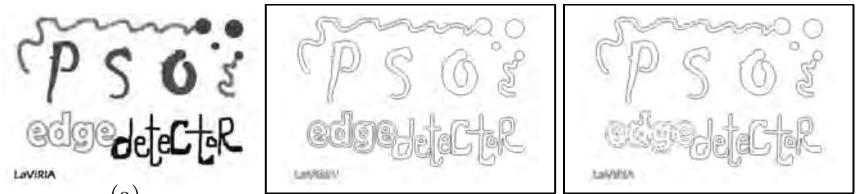
The F similarity metric combines P and R values in a single figure of merit, that takes values $F = [0, 1]$. A value 1 represents that both images are identical, and a score of 0 implies no matching edge pixels in both images. F value is computed as follows:

$$F = \frac{1}{2}\frac{PR}{P + R} \tag{10}$$

3 Implementation

We have run a PSO algorithm that uses as genes the complex coefficients of the spatial filter mask and the thresholding level. We have developed two test cases, both of them use a single training image S_1 (shown in Figure 2), but changing the process to obtain the image to be used as reference:

I A Canny-emulation filter, where a Matlab Canny operator is applied to S_1 in order to get a reference image, namely R_1^1.
II A Sobel-emulation filter, where a Matlab Sobel operator is applied to S_1 in order to get a reference image, namely R_1^2.

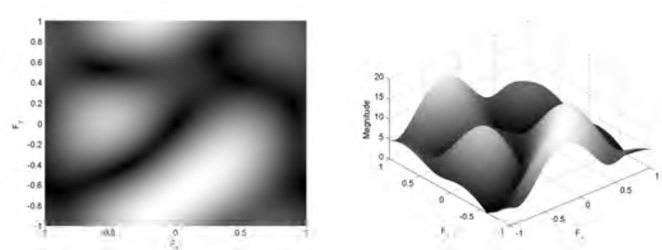


(a) (b) (c)

Fig. 2. (a) Original training image S_1, (b) the training references R_1^1 used to evolve the edge operator in case I, and (c) the training references R_1^2 used to evolve the edge operator in case II.

$$B_1^* = \begin{bmatrix} -1.9 + j\,1.0 & 2.7 - j\,1.4 & 2.5 - j\,0.7 \\ -1.1 + j\,1.0 & -0.8 + j\,2.5 & 1.6 + j\,2.8 \\ -2.9 - j\,2.0 & -0.8 - j\,2.3 & 0.9 - j\,1.0 \end{bmatrix}$$

(a)



(b)

Fig. 3. Optimal edge operator found in case I.

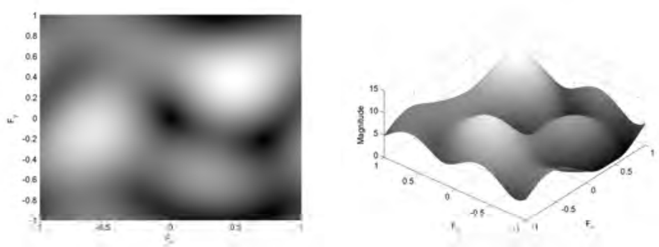
For each of these cases, an optimal edge filter, namely B_1^* and B_2^* were obtained for the case I and case II, respectively.

The coefficients of the spatial filter B_1^* are shown in Figure 3(a). Figure 3(b) is a graph of the frequency response of B_1^*. The optimal thresholding constant found by the PSO algorithm for this case was $T = 70.0$.

In the case II, the optimal set of coefficients of the spatial filter B_2^* are shown in Figure 4(a). In Figure 4(b), we can see a graph of the frequency response of B_2^*. The optimal thresholding constant found by the PSO algorithm for this case was $T = 47.0$.

$$B_2^* = \begin{bmatrix} 0.4 + j\,2.2 & 0.6 + j\,3.0 & 2.1 - j\,1.6 \\ -2.2 + j\,2.7 & 0.3 - j\,2.8 & 1.6 - j\,1.3 \\ 0.5 + j\,1.2 & -1.8 - j\,2.9 & -1.3 - j\,0.6 \end{bmatrix}$$

(a)



(b)

Fig. 4. Optimal edge operator found in case II.

4 Test and Results

We have applied the optimal edge detectors found in both cases (case I and case II) to a set I of test images, $I = \{I_1, \ldots, I_5\}$ selected from the Berkeley database. We have also used two sets of ground-truth images: U^1 (obtained from the application of a Matlab Canny edge operator) for test case I and U^2 (obtained from the application of a Matlab Sobel edge operator) for test case II.

Table 1 shows a qualitative summary of the results of the application of the optimal B_1^* and B_2^* edge operators and the expected results of using an optimal Matlab edge operator. As we can see there, results of the PSO-designed edge detectors are similar in appearance to the ground truth images. In order to compare quantitatively the results, Table 2 shows the F measures between images in the second and third columns in Table Qualitative and the fourth and fifth columns of the same table. We can observe that F measures are better for the Spbel like edge operator. That could be explained because Canny is not typically computed as a spatial filter. Conversely, Sobel was originally proposed as a spatial filter

These results were all obtained using 20 particles and 20 generations for the particle swarm evolution. Given this, we expect to improve F measures if we augment any of these PSO configuration parameters in further experiments.

5 Conclusions and Perspectives

We have presented an approach to obtain custom edge detectors designed using a PSO algorithm. The PSO algorithm optimizes simultaneously the parameters

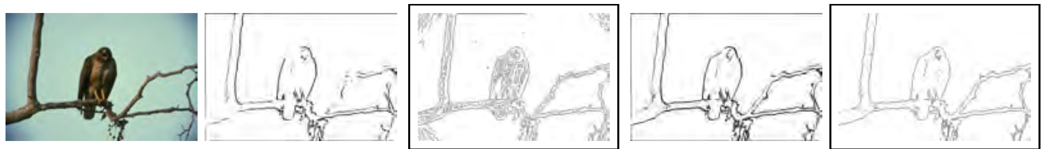
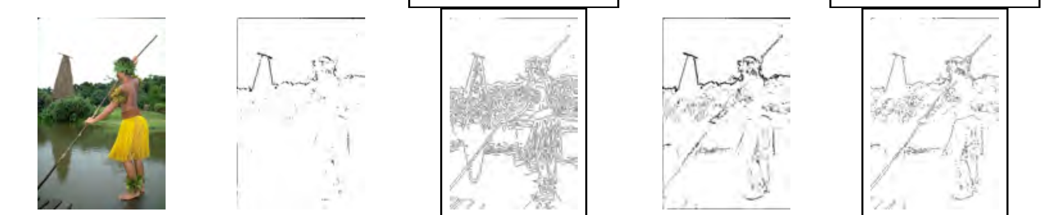
k	I_k	$I_k \otimes B_1^*$	U_k^1	$I_n \otimes B_2^*$	U_k^2
1					
2					
3					
4					
5					

Table 1. Qualitative results of the experimentation on a set of test images.

Test Image	$F(I_k, U_k^1)$	$F(I_k, U_k^2)$
I_1	0.399	0.602
I_2	0.139	0.580
I_3	0.115	0.560
I_4	0.185	0.504
I_5	0.220	0.585

Table 2. Results of the comparison between results images and their respective ground truth using the F metrics.

of a spatial filter and a thresholding step. We have shown the usefulness of the approach to emulate the response of a Canny and a Sobel edge operators. In both cases, qualitative and quantitative tests have been performed to validate our approach. Preliminary results show to be promising and we expect to work in the near future in tuning the PSO to improve its performance evaluation. The approach presented here could be easily extended for other image processing and computer tasks.

Acknowledgments

Ezequiel Martinez-Ayala gratefully acknowledges Mexico's CONACYT for the financial support through the scholarship 329514/229785.

References

1. Alipoor, M., Imandoost, S., Haddadnia, J.: Designing edge detection filters using Particle Swarm Optimization. In: Proc of the 18th Iranian Conference on Electrical Engineering (ICEE). pp. 548 –552 (May 2010)
2. Hongmei, T., Cuixia, W., Liying, H., Xia, W.: Image segmentation based on improved PSO. In: Proc. of the 2010 Int. Conf. On Computer and Communication Technologies in Agriculture Engineering (CCTAE). vol. 3, pp. 191 –194 (2010)
3. Kennedy, J., Eberhart., R.: Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA (2001)
4. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision. vol. 2, pp. 416–423 (July 2001)
5. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. on. Pattern Analysis and Machine Intelligence 26(5), 530–549 (2004)
6. Setayesh, M., Zhang, M., Johnston, M.: A new homogeneity-based approach to edge detection using PSO. In: Proc. of 24th Int. Conf. on Image and Vision Computing IVCNZ '09. pp. 231 –236. New Zealand. (Nov 2009)
7. Wei, K., Zhang, T., Shen, X., Liu, J.: An improved threshold selection algorithm based on Particle Swarm Optimization for image segmentation. In: Proc. of 3rd Int. Conf. onNatural Computation, 2007. ICNC 2007. vol. 5, pp. 591 –594 (Aug 2007)
8. Ye, Z., Chen, H., Liu, W., Zhang, J.: Automatic threshold selection based on Particle Swarm Optimization algorithm. In: Proc. of the 2008 Int. Conf. on Intelligent Computation Technology and Automation (ICICTA). vol. 1, pp. 36 –39 (Oct 2008)
9. Zhang, R., Liu, J.: Underwater image segmentation with maximum entropy based on Particle Swarm Optimization (PSO). In: First Int. Multi-Symposiums on Computer and Computational Sciences, 2006. IMSCCS '06. vol. 2, pp. 360 –636 (2006)
10. Zheng, L., Pan, Q., Li, G., Liang, J.: Improvement of grayscale image segmentation based on PSO algorithm. In: Proc. of the Fourth International Conf. on Computer Sciences and Convergence Information Technology, ICCIT '09. pp. 442 –446 (Nov 2009)