Expert Systems with Applications 39 (2012) 12407–12417

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An efficient method for segmentation of images based on fractional calculus and natural selection

Pedram Ghamisi [a,*], Micael S. Couceiro [b,c], Jón Atli Benediktsson [d], Nuno M.F. Ferreira [c,e]

[a] Geodesy & Geomatics Engineering Faculty, K. N. Toosi University of Technology, Tehran, Iran
[b] Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal
[c] RoboCorp at the Electrical Engineering Department, Engineering Institute of Coimbra, Rua Pedro Nunes – Quinta da Nora, 3030-199 Coimbra, Portugal
[d] Faculty of Electrical and Computer Engineering, University of Iceland, Saemundargotu 2, 101 Reykjavik, Iceland
[e] GECAD – Knowledge Engineering and Decision Support Research Center Institute of Engineering, Polytechnic of Porto (ISEP/IPP) Porto, Portugal

ARTICLE INFO

Keywords:
Multilevel segmentation
Swarm Optimization
Image processing

ABSTRACT

Image segmentation has been widely used in document image analysis for extraction of printed characters, map processing in order to find lines, legends, and characters, topological features extraction for extraction of geographical information, and quality inspection of materials where defective parts must be delineated among many other applications. In image analysis, the efficient segmentation of images into meaningful objects is important for classification and object recognition. This paper presents two novel methods for segmentation of images based on the *Fractional-Order Darwinian Particle Swarm Optimization* (*FODPSO*) and *Darwinian Particle Swarm Optimization* (*DPSO*) for determining the *n*-1 optimal *n*-level threshold on a given image. The efficiency of the proposed methods is compared with other well-known thresholding segmentation methods. Experimental results show that the proposed methods perform better than other methods when considering a number of different measures.

1. Introduction

Image segmentation is the process of partitioning a digital image into multiple regions. In other words, image segmentation could assign a label to each pixel in the image such that pixels with the same label share certain visual characteristics. These objects contain more information than individual pixels since the interpretation of images based on objects is more meaningful than that based on individual pixels. Image segmentation is considered as an important basic task in the analysis and understanding of images, thus being widely used for further image processing purposes such as classification and object recognition (Sezgin & Sankur, 2004).

Image segmentation can be classified into four different types including texture analysis based methods, histogram thresholding based methods, clustering based methods and region based split and merging methods (Brink, 1995). One of the most common methods for the segmentation of images is the thresholding method, which is commonly used for segmentation of an image into two or more clusters (Kulkarni & Venayagamoorthy, 2010).

Thresholding techniques can be divided into two different types: optimal thresholding methods (Kapur, Sahoo, & Wong, 1985; Kittler & Illingworth, 1986; Otsu, 1979; Pun, 1980; Pun, 1981) and property-based thresholding methods (Lim & Lee, 1990; Tsai, 1995 & Yin & Chen, 1993). The former group search for the optimal thresholds which make the thresholded classes on the histogram reach the desired characteristics. Usually, it is made by optimizing an objective function. The latter group detects the thresholds by measuring some selected property of the histogram. Property-based thresholding methods are fast, which make them suitable for the case of multilevel thresholding. However the number of thresholds is hard to determine and needs to be specified in advance.

Several algorithms have been proposed in literature that addressed the issue of optimal thresholding (Brink, 1995; Cheng, Chen, & Li, 1998; Huang & Wang, 1995; Hu, Hou, & Nowinski, 2006; Kapur et al., 1985; Li, Zhao, & Cheng, 1995; Otsu, 1979; Pun, 1980; Saha & Udupa, 2001; Tobias & Seara, 2002; Yin & Chen, 1993). While many of them address the issue of bi-level thresholding, others have considered the multi-level problem. The problem of bi-level thresholding is reduced to an optimization problem to probe for the threshold t that maximizes the σ_B^2 and minimizes σ_W^2 (Kulkarni & Venayagamoorthy, 2010). For two level thresholding, the problem is solved by finding T^* which satisfies max

* Corresponding author. Address: No. 11, Fotouhi st., Farabi st., Ardakani st., Guyabadi st., Zafar st., Shariati st. Tehran, Iran Postal code: 1916759333. Tel.: +98 912 207 82 63.

E-mail addresses: p.ghamisi@gmail.com (P. Ghamisi), micaelcouceiro@isr.uc.pt, micael@isec.pt (M.S. Couceiro), benedikt@hi.is (J.A. Benediktsson), nunomig@isec.pt (N.M.F. Ferreira).

0957-4174/$ - see front matter
http://dx.doi.org/10.1016/j.eswa.2012.04.078

$(\sigma_B^2(T^*))$ where $0 \leqslant T^* < L$ and L is the maximum intensity value. This problem could be extended to n-level thresholding through satisfying max σ_B^2 $(T_1^*, T_2^*, \ldots, T_{n-1}^*)$ that $0 \leqslant T_1^* < T_2^* < \cdots < T_{n-1}^* < L$. One way for finding the optimal set of thresholds is the exhaustive search method. The exhaustive search method based on the Otsu criterion (Otsu, 1979) is simple, but it has a disadvantage that it is computationally expensive (Kulkarni & Venayagamoorthy, 2010). Exhaustive search for $n-1$ optimal thresholds involves evaluations of fitness of $n(L-n+1)^{n-1}$ combinations of thresholds (Kulkarni & Venayagamoorthy, 2010) so this method is not suitable from a computational cost point of view. The task of determining $n-1$ optimal thresholds for n-level image thresholding could be formulated as a multidimensional optimization problem.

Alternative to the Otsu method, several biologically inspired algorithms have been explored in image segmentation (Fogel, 2000; Kulkarni & Venayagamoorthy, 2010; Lai & Tseng, 2004; Yin, 1999). Bio-inspired algorithms have been used in situations where conventional optimization techniques cannot find a satisfactory solution, for example when the function to be optimized is discontinuous, non-differentiable, and/or presents too many nonlinearly related parameters (Floreano & Mattiussi, 2008). The Particle Swarm Optimization (PSO) is a machine-learning technique loosely inspired by birds flocking in search of food (Kennedy & Eberhart, 1995). It basically consists of a number of particles that collectively move in the search space (*e.g.*, pixels of the image) in search of the global optimum (*e.g.*, maximizing the between-class variance of the distribution of intensity levels in the given image). However, a general problem with the PSO and other optimization algorithms is that of becoming trapped in a local optimum, such that it may work in some problems but may fail on others (Couceiro, Ferreira, & Machado, 2011).

The Darwinian Particle Swarm Optimization (*DPSO*) was formulated by Tillett et al. in 2005 (Tillett, Rao, Sahin, Rao, & Brockport, 2005) in search of a better model of natural selection using the *PSO* algorithm. In this algorithm, multiple swarms of test solutions performing just like an ordinary *PSO* may exist at any time with rules governing the collection of swarms that are designed to simulate natural selection. More recently, an extension of the *DPSO* using fractional calculus to control the convergence rate of the algorithm was presented by Couceiro et al. in 2011 (Couceiro et al., 2011), being denoted as fractional-order *DPSO* (*FODPSO*). The novel algorithm was successfully compared with both the fractional-order *PSO* from Pires, Machado, Oliveira, Cunha, and Mendes (2010) and the traditional *DPSO*.

Significant progress has been made in the creative inspiration of bio-inspired computer algorithms applied to optimization, estimation, control and many others through the application of principles derived from the study of biology (Floreano & Mattiussi, 2008). Santana, Alves, Correia, and Barata (2010) presented a swarm-based model for trail detection in real-time. Experimental results on a large dataset revealed the ability of the model to produce a success rate of 91% using a 20 Hz camera with a resolution of 640×480 that was carried through a scenario at an approximate speed of 1 m s^{-1}. The authors in Kulkarni and Venayagamoorthy (2010) compared the *PSO* and Bacteria Foraging algorithm (*BF*) with the Otsu method to determine the optimal threshold level for the deployment of sensor nodes. It should be noted that all methods were run offline and the *PSO* presented a superior performance when compared to the Otsu and the *BF*. Omran (2004) presented the application of the *PSO* to the field of pattern recognition and image processing. He introduced a clustering algorithm based on *PSO*. Further, he developed a dynamic clustering algorithm that could find the "optimum" number of clusters in a dataset with minimum user interference. Sathya and Kayalvizhi (2010) proposed a multilevel thresholding method based on *PSO* and compared their method with *GA*-based thresholding method. Results showed that the *PSO*-based image segmentation executed faster and was more stable than *GA*.

This paper mainly focuses on using one of the best performing *PSO* main variants (*cf.*, (Couceiro, Luz, Figueiredo, Ferreira, & Dias, 2010; Couceiro et al., 2011), created by Tillett et al. (2005)), denoted as *DPSO*, and the recently fractional-order extension, denoted as *FODPSO* (Couceiro et al., 2011). This is the first work to verify and apply the *FODPSO* and *DPSO* to multilevel segmentation. Bearing this idea in mind, the problem formulation of image n-level thresholding is presented in the following sub-sections. Section 2 presents a brief review of particle swarm algorithms, focusing on the strengths and weaknesses of the traditional *PSO*, the *DPSO* and the *FODPSO*. In Section 3, several images used to compare the *PSO*-based image segmentation variants with other commonly used algorithm such as genetic algorithms (*GA*) and *BF*. Finally, Section 4 outlines the main conclusions.

1.1. Image thresholding

Multilevel segmentation techniques provide an efficient way to perform image analysis. However, the automatic selection of a robust optimum n-level threshold has remained a challenge in image segmentation. This section presents a more precise formulation of the problem, introducing some basic notation.

Let there be L intensity levels in each *RGB* (red-green–blue) component of a given image and these levels are in the range $\{0,1,2,\ldots,L-1\}$. Then one can define:

$$p_i^C = \frac{h_i^C}{N}, \quad \sum_{\substack{i=1 \\ C=\{R,G,B\}}}^{N} p_i^C = 1, \tag{1}$$

where i represents a specific intensity level, *i.e.*, $0 \leqslant i \leqslant L-1$,$C$ represents the component of the image, *i.e.*, $C = \{R,G,B\}$,N represents the total number of pixels in the image and h_i^C denotes the number of pixels for the corresponding intensity level i in the component C. In other words, h_i^C represents an image histogram for each component C, which can be normalized and regarded as the probability distribution p_i^C. The total mean (*i.e.*, combined mean) of each component of the image can be easily calculated as:

$$\mu_T^C = \sum_{\substack{i=1 \\ C=\{R,G,B\}}}^{L} i p_i^C. \tag{2}$$

The 2-level thresholding can be extended to generic n-level thresholding in which $n-1$ threshold levels $t_j^C, j = 1, \ldots, n-1$, are necessary and where the operation is performed as expressed below:

$$F^C(x,y) = \begin{cases} 0, & f^C(x,y) \leqslant t_1^C \\ \frac{1}{2}\left(t_1^C + t_2^C\right), & t_1^C < f^C(x,y) \leqslant t_2^C \\ \vdots & \vdots \\ \frac{1}{2}\left(t_{n-2}^C + t_{n-1}^C\right), & t_{n-2}^C < f^C(x,y) \leqslant t_{n-1}^C \\ L, & f^C(x,y) > t_{n-1}^C \end{cases} \tag{3}$$

where x and y are the width (W) and height (H) pixel of the image of size $H \times W$ denoted by $f^C(x,y)$ with L intensity levels in each *RGB* component. In this situation, the pixels of a given image will be divided into n classes $D_1^C, \ldots, D_n^C$, which may represent multiple objects or even specific features on such objects (*e.g.*, topological features).

The simplest and computationally most efficient method of obtaining the optimal threshold is the one that maximizes the between-class variance which can be generally defined by: