Pattern Recognition Letters 74 (2016) 46–52

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Learning automata for image segmentation☆

Qian Sang[a], Zongli Lin[b], Scott T. Acton[b,*]

[a] Algorithm Research, AOL Platforms R&D, Palo Alto, CA 94306, USA
[b] Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904-4743, USA

A R T I C L E I N F O

Article history:
Received 29 May 2015
Available online 11 February 2016

Keywords:
Image analysis
Image segmentation
Automata

A B S T R A C T

A method is proposed seeking thresholds for segmentation of grayscale images. The normalized image histogram is modeled as a Gaussian mixture, and the parameters associated with the Gaussian components are estimated iteratively with a set of learning automata. To reduce the parameter search space, the number of major components in the image and their associated parameter ranges are first specified using some desired properties of the Gaussian distribution. Thresholds are chosen based on the Gaussian parameter estimates after convergence. Illustrative examples are provided to demonstrate the learning process and the effectiveness of the proposed segmentation scheme.

1. Introduction

Image segmentation is a fundamental and ubiquitous task in higher-level image analysis and processing applications such as object detection and classification and image retrieval. The quality of segmentation has a direct impact on the effectiveness and validity of these higher-level tasks. The objective of image segmentation is to partition an image into meaningful regions corresponding to objects of interest. Thresholding is an effective and commonly used segmentation method due to its simplicity. It is based on the assumption that different objects can be distinguished solely based on the image histogram. When more than one object with distinctive features exists in the image, thresholds are sought for an appropriate segmentation of each object out of the background.

A wide variety of threshold selection techniques for image segmentation exist in the literature. Conventional methods pick a threshold that minimizes the overlap or maximizes the variance between two adjacent histogram clusters based on certain statistical metrics [11,19]. More advanced techniques employ intelligent concepts for segmentation such as the support vector machine [7], particle swarm optimization [28], and information entropy [18,21,24]. An alternative threshold selection technique views the image histogram as a probability distribution function (PDF) and models it as a finite mixture of probability distributions [6]. Among various distribution mixture models, the Gaussian mixture model (GMM) has been well studied and widely used, due to its simple representation with each univariate Gaussian distribution requiring only two parameters (the mean and the standard deviation) and its desirable characteristics, such as symmetry, isotropy, and unimodality [12,35].

In the context of a GMM, threshold selection becomes a three-step process: (i) detect the number of Gaussian components from the normalized histogram of an image; (ii) estimate the Gaussian distribution parameters (mean, standard deviation, and weight) by fitting the mixture to best approximate the normalized histogram, an optimization problem; and (iii) calculate thresholds based on the estimated Gaussian parameters. Schemes to estimate the number of Gaussian components and their associated parameters, either iterative or non-iterative, have been proposed for multi-level thresholding [3,4,6,27]. Since the tails of the Gaussian components in the mixture model are in general overlapping, estimates from non-iterative schemes without refinement are inadequate [4]. Because of the large parameter search space (especially for higher-dimensional cases), estimation of Gaussian parameters usually employs gradient-based numerical search methods [20]. However, these methods can easily be trapped at local extrema in the optimization space, and reply heavily on the accuracy of the initial estimate [12]. Alternative iterative estimation methods such as the Expectation-Maximization (EM) algorithms [34] suffer from additional problems in practical implementation, e.g., slow convergence and sensitivity to initial conditions [31].

In this paper, we employ a learning-based optimization approach utilizing learning automata [29] for estimating Gaussian mixture parameters. Different learning automaton algorithms are distinguished by the ways their PDFs are updated. They find applications in signal processing [14], feedback control systems [13,33], power systems [30], and image processing [2,8,22,25]. The

☆ This paper has been recommended for acceptance by Prof. M. Couprie.

* Corresponding author. Tel.: +1 434 982 2003; fax: +1 434 924 8818.
E-mail addresses: qs2v@virginia.edu (Q. Sang), zl5y@virginia.edu (Z. Lin), acton@virginia.edu (S.T. Acton).

http://dx.doi.org/10.1016/j.patrec.2015.12.004

randomness in the generation of parameter estimates provides learning automaton algorithms with obvious advantages over the gradient-based and EM algorithms. To be specific, convergence properties are independent of initial conditions, global extrema can be achieved over the learning process [1], and the speed of convergence is faster [29], especially with an increasing number of parameters to be estimated.

We approach the Gaussian parameter estimation problem with the so-called continuous action reinforcement learning automata (CARLA) [14] algorithm, in the same vein as in [8]. In particular, the output of each automaton corresponds to one specific Gaussian parameter to be estimated. Their combination at each stage provides a fitted Gaussian mixture model of the original image histogram. The match between the two is evaluated based on some metric, e.g., the average mean square error, and is returned as a reinforcement signal, to be applied as the input to each automaton. Inside a learning automaton, a probability distribution function (PDF) is defined over the parameter search range, representing the desirability of each specific value, and is updated as learning proceeds based on the reinforcement signal. A parameter estimate (an automaton output) is randomly generated for the next stage based on the current probability distribution. However, different from [8,14], we employ some desired properties of the Gaussian distribution to pre-process an image histogram to detect the number of major components and shrink parameter search ranges associated with each component. This significantly expedites convergence of the iterative estimation process.

The rest of the paper is organized as follows. Section 2 provides a detailed description of the learning-based thresholding method. Illustrative examples are given in Section 3 for the segmentation of a remotely sensed image to illustrate the effectiveness of the proposed method. Conclusions and some future work are given at the end of the paper.

2. A learning-based thresholding method

In this section, we consider the multi-level thresholding problem for an image that has a normalized histogram modeled as a mixture of Gaussians, the so-called Gaussian mixture model (GMM). Each Gaussian component in this case is assumed to correspond to an object (or a group of objects with similar pixel grayscale level distributions) of interest. Two key issues are to be addressed: (1) the number of components in the GMM needs to be determined; and (2) parameter estimates of each Gaussian component are required for threshold computations. When individual Gaussian components are well separated from each other, peaks and valleys in the GMM may be used to detect the presence of Gaussian components. This method is not valid, however, when a relatively smaller component is close to but not dominated by a larger one. We adopt the method used in [4]: a Gaussian component is detected with a pair of zero crossings in the second order difference of the GMM. In an ideal Gaussian density distribution with a mean μ and a standard deviation σ, this pair of zero crossings occur at the inflection points of the Gaussian, i.e., $\mu \pm \sigma$. The pair does not disappear even when two components are close to each other.

Since the tails of the Gaussian components overlap with each other to a large extent in general, the locations of zero crossings cannot be used to precisely extract the individual component. We adopt an iterative learning procedure, in which locations of the zero crossings are used to initialize the search range for a given parameter, and a learning automaton [1] is employed to progressively refine the estimate.

The Gaussian component detection and localization procedure outlined above is essentially a two-step process, as shown in Fig. 1. Detailed descriptions of each major step are provided next.

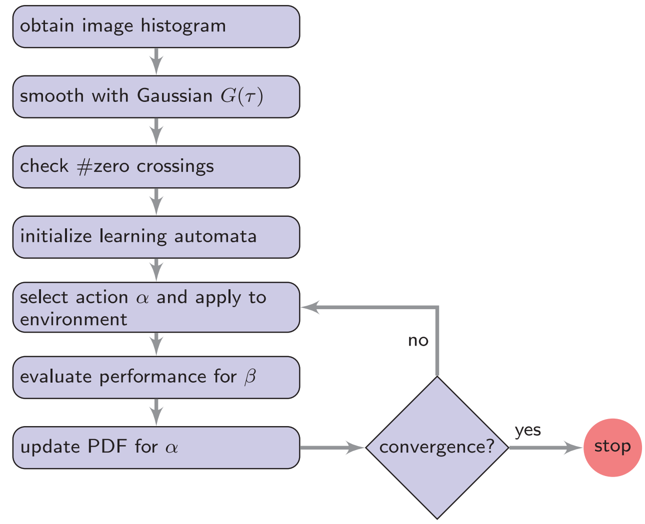

Fig. 1. Gaussian component detection and parameter estimation with learning automata.

2.1. Gaussian component detection

The histogram of an image provides statistical information on its pixels with different grayscale levels. It is first normalized to be viewed as a probability density function $h(x)$, $x \in [0, L-1]$, where L is the number of different grayscale levels used in the image representation. The normalized histogram $h(x)$ is then modeled as a Gaussian mixture $G(x)$. To avoid over-segmentation, the histogram for a real image is first smoothed by a Gausssian kernel with a chosen standard deviation τ

$$G_\tau(x) = \frac{1}{\sqrt{2\pi}\,\tau} \exp\left[-\frac{x^2}{2\tau^2}\right] \tag{1}$$

to obtain a smoothed normalized histogram $h_\tau(x)$ with

$$h_\tau(x) = G_\tau(x) * h(x) \tag{2}$$

where "$*$" denotes the convolution operation.

Zero crossings of the second order derivative approximation signal for $h_\tau(x)$ [9] can then be detected. Note that the zero crossings should appear in pairs with the sign of the second order difference of $h_\tau(x)$ changing from positive to negative at the left crossing point, and from negative to positive at the right crossing point. These inflection points occur at $\mu \pm \sigma$ for an ideal Gaussian $G(\mu, \sigma)$, and provide initial estimates of its mean and standard deviation.

Each pair of inflection points determine the presence of a Gaussian component. Suppose there are K pairs of inflection points detected from $h_\tau(x)$. The Gaussian mixture model of $h_\tau(x)$ is expressed as

$$G(x) = \sum_{i=1}^{K} \frac{p_i}{\sqrt{2\pi}\,\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right] \tag{3}$$

where μ_i, σ_i, and p_i are the mean, the standard deviation, and the weight of the ith component in the mixture, respectively. Noting that

$$\sum_{i=1}^{K} p_i = 1 \tag{4}$$

we have in total $3K-1$ independent parameters to be estimated by the learning procedure.

Initialization of learning process. The PDF for the estimation of a Gaussian parameter is usually initialized to be a uniform