A parallel adaptive segmentation method based on SOM and GPU with application to MRI image processing

Ailing De, Yuan Zhang, Chengan Guo*

School of Information and Communication Engineering, Dalian University of Technology, Dalian 116023, China

ARTICLE INFO

Article history:
Received 22 April 2015
Received in revised form
10 August 2015
Accepted 27 October 2015
Available online 8 March 2016

Keywords:
Image segmentation
Parallel processing
SOM neural network
Vector quantization
GPU

1. Introduction

Image segmentation has been a very active research topic over several decades due to its essential role played in image analysis and computer vision. Many segmentation methods have been proposed in the existing literature (e.g., see [1–7]). The aim of image segmentation is to divide an image into some meaningful parts for further uses. There are two basic requirements for an image segmentation method: one is the segmentation accuracy that is desired to be as accurate as possible, and the other is the processing speed that is to be as fast as in need for applications. Usually the two requirements are contradictory each other. In order to meet the first requirement, both the gray value information and the spatial/structural information of the image must be effectively exploited by the segmentation algorithm, thus causing the algorithm more sophisticated. Obviously, the more complex of the algorithm, the more heavy its computational load will get. In addition, due to a large amount of data to be processed in an image, it makes real-time processing in image segmentation much more difficult. Recently, progresses have been made in developing parallel segmentation algorithms based on multi-core processors or graphic processing units (GPU), which can speed up the segmentation process significantly (e.g., see [8–11]). As a matter of fact, nowadays GPUs have emerged as a powerful, low-cost, and general purpose parallel computation platform and many successful works have been conducted in the field of signal/image processing and intelligent computation (e.g., [11–14]). With the rapid developing of GPU and multi-core technologies, it is a very promising direction to develop GPU-based parallel algorithms for solving the real-time processing bottleneck in the computation field. However, research in this area is far from over and lots of works need to do since most of the existing methods in the field are serial ones that cannot be directly implemented on GPUs in parallel. Moreover, different problem usually requires using different parallel scheme that suits well only to its own. For the segmentation problem, for example, hundreds of segmentation methods have been proposed so far, including histogram thresholding, clustering, region-based, contour-based, energy-based, and miscellaneous algorithms [15]. Each kind of the methods has its own characteristics in methodology since in usual a segmentation method is designed in according to its own problems concerned. In fact, there has been no universal segmentation method yet that is suitable for all the segmentation problems, which perhaps is the reason why so many segmentation algorithms have been proposed so far. As a result, there is also no universal parallel scheme that can accommodate itself to all the existing segmentation approaches. Therefore, it still has a long way to go in the field of parallel segmentations.

In [7], we proposed an adaptive vector quantization method for image segmentation, in which the self-organizing map (SOM) network...
and vector quantization (VQ) method are integrated together and applied to segmenting the human brain MRI images with excellent performance. As pointed in [7], however, the computational complexity of the method is very high and it is in need to accelerate the processing speed. In [11], we presented parts of the parallel algorithms for the SOM-based VQ segmentation method. In this paper we present a complete set of parallel algorithms for the method and implement them on a GPU platform using parallel programming language and give more extensive experimental results and analyses.

The sequel of the paper is organized as follows: Section 2 describes the proposed parallel algorithms, including the overall parallel scheme, the parallel algorithm for vector representation of images, the parallel classification of edge and non-edge pattern vectors, the parallel training of SOM network, the parallel quantization of the non-edge pattern vectors, and the parallel classification of the edge pattern pixels. Section 3 presents the parallel algorithms for estimating the segment number of the image being processed adaptively. Experimental results with applications to MRI image segmentation are given in Section 4. Section 5 summarizes the work and points out the possible direction of the paper.

2. Parallel algorithms for SOM-based VQ segmentation method

In this section, after a brief summary on the original SOM-based VQ segmentation method proposed in [7], we present an overall parallel scheme for the approach at first, then describe the corresponding parallel algorithms and concrete implementation steps for the scheme.

2.1. The overall parallel scheme of the segmentation method

The original SOM-based VQ segmentation method of [7] includes the following computational processes:

1. Divide the image to be segmented into sub-blocks, as illustrated in Fig. 1, each sub-block consisting of \( n \times n \) pixels and being represented with a vector of \( n^2 \) elements.
2. Classify the sub-block vectors into two patterns, known as the edge pattern and non-edge pattern, by using the edge detection algorithm based on the wavelet modulus maximum edge detection.
3. Train an SOM neural network by using the non-edge pattern vectors as the training samples.
4. Cluster (quantize) the non-edge pattern vectors into \( K_c \) classes by using the trained SOM network with VQ method.
5. Classify the pixels of the edge pattern sub-blocks into the clusters obtained in the VQ procedure.
6. Estimate the segment number \( K_c \) with an adaptive search method by minimizing the ratio of the traces of the within-class scatter matrix and between-class scatter matrix obtained in the clustering process.

In this paper, we design parallel algorithms for all of the above computation processes and implement the parallel algorithms on a GPU platform with parallel programming language OpenCL [16,17]. The overall parallel scheme is illustrated in Fig. 2 that includes 6 parallel computation modules corresponding to the above 6 processes. The parallel algorithms will be described in detail in subsequent sections.

2.2. Parallel algorithm for sub-block dividing and vector representation of an image

The first step in the VQ segmentation method [7] is to divide the image being processed into small sub-blocks and represent them with vectors. In this section, we design a simple parallel algorithm to implement this operation.

As shown in Fig. 1, suppose that the image to be segmented is denoted by \([f(i,j)]_{MxN}\) and we want to represent the image with \( N_v \) vectors, \( \{X(k); k=1, ..., N_v\} \), in which each vector is constructed with the pixels of a sub-block \([f(i, j)]_{m\times n}\) of the image, where \( M \) and \( N \) are the height and width of the image respectively, \( n \) is the height or width of the sub-block with \( n \ll \min(M, N) \), and the number of the vectors \( N_v = [MN/n^2] \).

The parallel algorithm for realizing the sub-block dividing and vector representation process is given below.

- **Parallel Algorithm 2.1**

\[
\begin{align*}
&\text{begin} \\
&\quad \text{for } k = 1 \text{ to } N_v \text{ do in parallel} \\
&\quad \quad (i) \text{ find the location of the } k\text{-th sub-block } [f(i, j)]_{m\times n}, \text{ in } [f(i, j)]_{MxN}; \\
&\quad \quad (ii) \text{ construct vector } X(k) \text{ with pixels of the } k\text{-th sub-block by } \\
&\quad \quad \quad X(k) = [f(i_1, j_1), ..., f(i_{m}, j_{m}), ..., f(i_{n,1}, j_{n,1}), ..., f(i_{n, m}, j_{n, m})]'; \\
&\quad \quad \text{end for} \\
&\text{end}
\end{align*}
\]

Obviously, for the above parallel algorithm, the parallel degree is \( N_v \) and the theoretical speedup ratio is also \( N_v \), compared with the original serial algorithm. Usually \( N_v \) is very large since \( n = \min(M, N) \). For a 256 \( \times \) 256 image, for example, a significant speedup ratio for the parallel algorithm can be obtained, in which \( N_v = 4096 \) when the sub-block size is set to \( n = 4 \).

2.3. Parallel classification of the vectors into edge and non-edge patterns

The second processing module of the VQ segmentation method is to classify the vectors (sub-blocks) into two patterns, the edge pattern and non-edge pattern, by using the wavelet modulus maximum edge detection method [18,19]. The classification method proposed in [7] uses the partial derivatives of the two-dimensional Gaussian function as wavelet functions and involves the following 6 computation steps:

- **Step 1**: use the two wavelet functions as filters to filter the image by the following convolutions:

\[
\begin{align*}
W^1 f(x, y) &= f(x, y) \otimes \phi_1^2(x, y), \\
W^2 f(x, y) &= f(x, y) \otimes \phi_2^2(x, y),
\end{align*}
\]

(1)

where \( f(x, y) \) denotes the image function, \( \phi_1^2(x, y) \) and \( \phi_2^2(x, y) \) are the wavelet kernel functions, \( \phi_2^2(x, y) = \frac{1}{16\sigma^2} e^{-(x^2 + y^2)/(16\sigma^2)} \) and \( \phi_1^2(x, y) = e^{-(x^2 + y^2)/(4\sigma^2)} \).

- **Step 2**: compute the modulus and angle parameters, \( M_2 f(x, y) \) and \( A_2 f(x, y) \), based on the above results:

\[
M_2 f(x, y) = \sqrt{|W^1 f(x, y)|^2 + |W^2 f(x, y)|^2}.
\]
دریافت فوری
متن کامل مقاله
امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات