

세포 외곽선 추출 알고리즘의 병렬화

이호석[†], 유숙현^{**}, 권희용^{***}

Parallelization of Cell Contour Line Extraction Algorithm

Ho Seok Lee[†], Suk Hyun Yu^{**}, Hee Yong Kwon^{***}

ABSTRACT

In this paper, a parallel cell contour line extraction algorithm using CUDA, which has no inner contour lines, is proposed. The contour of a cell is very important in a cell image analysis. It could be obtained by a conventional serial contour tracing algorithm or parallel morphology operation. However, the cell image has various damages in acquisition or dyeing process. They could be turn into several inner contours, which make a cell image analysis difficult. The proposed algorithm introduces a min-max coordinates table into each CUDA thread block, and removes the inner contour in parallel. It is 4.1 to 7.6 times faster than a conventional serial contour tracing algorithm.

Key words: Image Processing, Cell Image, Contour Line, Parallel Processing, CUDA

1. 서 론

최근 세포 영상에 대한 자동 분석이 의료 영상 분야에서 활발히 연구되고 있다. 이러한 세포 영상의 분석 결과는 의사의 진단 근거로 이용되기 때문에 높은 정확성이 요구되는데, 특히, 세포의 외곽선은 세포의 정보를 분석하는데 있어 매우 중요한 영향을 미친다. 이에 다양한 외곽선 추적 알고리즘(contour tracing algorithm)들이 연구되어 왔다.

외곽선 추적 알고리즘(contour tracing algorithm)은 경계선 추적 알고리즘(boundary following algorithm)이라고도 하며, 영상에서 특정한 물체의 외곽선을 추적하는 알고리즘이다. 외곽선 추적을 이용하여 배경 영상으로부터 물체를 분리할 수 있으며, 물체의 외곽선 길이를 알 수 있으므로 물체의 크기와 모양을 추정할 수 있다. 또한 외곽선을 구성하는 픽셀들의 방향 정보에 따라 물체의 선분 정보와 방향

변화 정보를 알 수 있으므로 물체의 특징 점을 찾아내는 경우에도 매우 유용하고, 물체를 영상에 표현하거나 묘사할 경우에도 이용된다[1].

외곽선 추적 알고리즘으로는 대상 영상을 관심영역과 비관심 영역으로 분류하고, 체인코드를 생성하여 경계선을 추출하는 방식의 간단한 경계선 추적자 알고리즘(SBF : simple boundary follower)[2]과 SBF의 비일관성 문제를 개선한 수정된 간단한 경계선 추적자 알고리즘(MSBF: modified simple boundary follower)[3], 8-방향성을 이용하여 외곽선을 추적해 나가는 방사형 탐색 알고리즘(radial sweep algorithm)[4]과 Theo Pavlidis 알고리즘[5] 등이 있다. 이 알고리즘들은 매우 간단하여 적용이 용이한 장점이 있으나, 응용 영역에 따라 유효하지 않은 경우가 존재하며, 순차적으로 알고리즘이 수행되기 때문에 영상의 크기가 증가하면, 실행속도가 늦어질 수 있다. 특히, 의료영상과 같은 고용량의 데이터가 사용

* Corresponding Author: Hee Yong Kwon, Address: (430-714) 22, Samdeok-ro 37 beon-gil, Manan-gu, Anyang-si, Gyeonggi-do, Korea, TEL: +82-31-467-0878, FAX: +82-31-463-1249, E-mail: hykwon@anyang.ac.kr

Receipt date: Aug. 10, 2015, Approval date: Aug. 20, 2015

[†] Dept of Computer Engineering, Anyang University (E-mail: gov676@nate.com)

^{**} Dept of Information & Communications Engineering, Anyang University (E-mail: shy@anyang.ac.kr)

^{***} Dept of Computer Engineering, Anyang University

되는 영상처리 시스템에 적용 시 이러한 문제는 더욱 심화될 수 있다.

본 연구와 유사하게 외곽선 추출 알고리즘을 의료 영상에 적용한 기존 연구로는 기도에 대한 내벽 윤곽선을 검출하거나[6], 외곽선 검출을 통하여 폐 영역을 분할하거나[7], 외곽선 추출 알고리즘을 이용해 세포의 수와 면적을 측정한 연구[8] 등이 다수 존재한다. 하지만, 이들 연구에서는 모두 영상의 색상차를 이용하거나, 문턱치(threshold) 값을 이용하여 순차적으로 외곽선을 탐색하였기 때문에 앞서 소개한 기존 연구들과 마찬가지로 고용량의 의료 영상 데이터에 대해서는 효율적이지 못하다. 따라서, 이러한 문제를 해결하기 위해서는 병렬처리가 필요한데, 외곽선 추출 알고리즘과 같이 추출과정에서 하나의 외곽점 추출 결과가 다음 외곽점 추출 결과에 영향을 미치는 경우, 경합조건(race condition)이 발생하여 병렬화를 시키기가 어렵다. 이런 문제점의 해결 방안으로서 에지 필터(edge filter)나 모폴로지(morphology)연산 등을 이용하여 외곽선을 추출하는 방법을 사용할 수도 있다[1]. 이 연산들은 연산 영역간 상호 의존성이 없는 영역 기반 연산이기 때문에 병렬화가 용이하다[9].

그러나 이 같은 알고리즘들은 원영상의 세포 내부에 공간이 있는 경우 기존의 외곽선 추적 알고리즘과는 달리 빈 공간이 그대로 내부 외곽선 형태로 나타난다. 내부 외곽선이란 객체에서 추출되는 외곽선중 가장 바깥쪽에 위치한 외곽선을 제외한 모든 외곽선을 말한다. 외곽선 정보를 이용하여 세포의 정보를 추출할 때, 내부 외곽선은 정확한 세포의 정보 분석을 어렵게 만드는 원인으로 작용한다.

Fig. 1은 세포의 다양한 형태에 따라 순차처리 시

스템과 병렬처리 시스템에서의 외곽선 추출 결과가 다른 것을 보이고 있다. Fig. 1(a)는 원 영상에 대한 이진영상이고, Fig. 1(b)는 순차처리 시스템에서 외곽선을 추출한 결과, Fig. 1(c)는 병렬처리 시스템에서 외곽선을 추출한 결과이다. Fig. 1의 (b)와 (c)에 원으로 표기한 중앙의 세 개의 세포를 보면 순차처리 시스템에서 나타나지 않았던 내부 외곽선이 병렬처리 시스템에서 나타난 것을 확인할 수 있다. Fig. 2는 외곽선 정보를 이용한 세포 정보 분석 작업들 중에서 세포의 장,단축 추출 결과를 나타낸 그림이다. Fig. 2의 A, B를 보면 단축의 끝이 최 외곽 픽셀이 아닌 것을 볼 수 있는데, 내부 외곽선이 존재함으로써 정확한 세포 정보 분석이 이루어지지 않은 것을 알 수 있다.

따라서, 빠르고 정확한 세포의 정보 분석을 위하여, 본 논문에서는 병렬로 외곽선을 추출하고 이때 발생하는 내부 외곽선을 병렬로 제거하는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 병렬처리 시스템에서 외곽선을 추출하기 위한 전처리 과정과 내부 외곽선을 제거하는 방법을 서술한다. 3장은 기존의 순차처리 외곽선 추적 알고리즘과 본 연구에서 제안하는 병렬처리 외곽선 추출 알고리즘에 대한 성능평가를 한다. 마지막으로, 4장에서는 결론을 제시하고, 향후 연구에 대해 살펴본다.

2. 병렬 세포 외곽선 추출

2.1 알고리즘 개요

본 논문에서 제안하는 병렬 세포 외곽선 추출 알고리즘은 크게 전처리 작업과 내부 외곽선 제거 작업으로 나누어진다. 전처리 작업은 병렬로 외곽선을 추

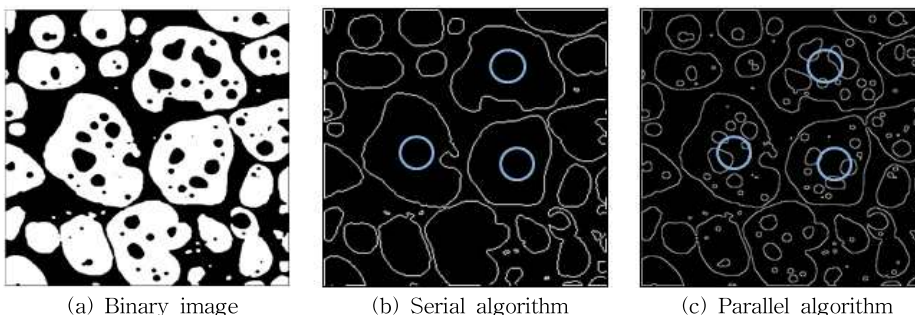


Fig. 1. Comparison between a serial algorithm and a parallel algorithm.

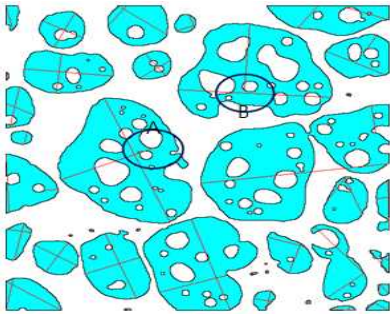


Fig. 2. The major axis and the minor axis of a cell.

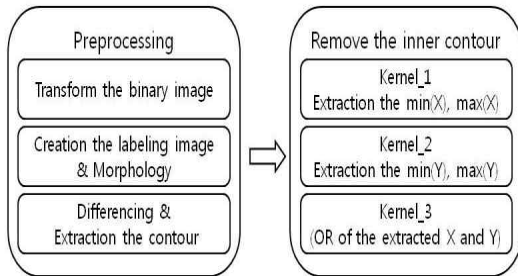


Fig. 3. Flowchart for parallel cell contour line extraction procedure.

출하는 단계로 입력영상을 이진영상으로 변환하고, 이진영상에 대하여 레이블링(labeling)을 수행한다. 잡음제거를 위하여 모폴로지(morphology)연산인 열림 연산을 수행 한 후, 레이블링 영상과 모폴로지 연산인 침식연산을 수행한 레이블링 영상과의 차를 통해 레이블링된 외곽선을 추출한다. 내부 외곽선 제거 단계에서는 객체별로 추출된 외곽선의 좌표를 비교하여 각 객체의 내부 외곽선을 제거한다. Fig. 3은 제안하는 병렬 세포 외곽선 추출 알고리즘의 전체적인 흐름도를 나타낸다.

2.2 선택 이진 영상 변환

영상처리에서 이진화(binartization)란 영상의 픽셀 값을 0 또는 255로 만드는 연산을 의미하며, 이는 픽셀의 속성을 배경(background)과 객체(object) 두 개의 그룹으로 나눔을 의미한다. 영상의 이진화는 다양한 영상처리 분야에서 사용되며, 특히 영상 내에 원하는 객체의 위치를 찾기 위한 전처리 과정으로 많이 사용된다. 이진화를 수행하기 위해서는 영상 내 모든 픽셀에 대하여 그레이스케일 값이 특정 값보다 크면 255로 바꾸고, 작으면 0으로 바꾸는 방법을 사용한다. 이때, 픽셀 값의 크기를 비교하는 대상이 되는 값을 임계값(threshold) 또는 문턱치라고 부른다. 임계값은 그레이스케일 범위인 0~255 사이의 정수 값을 사용한다. 이진화 과정을 수식으로 표현하면 다음과 같다[10].

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) \leq T \\ 255 & \text{if } f(x,y) > T \end{cases} \quad (1)$$

여기서 $f(x,y)$ 와 $g(x,y)$ 는 각각 입력영상과 출력영상을 의미하고, T 는 임계값을 의미한다.

영상 이진화에 사용되는 다양한 임계값 결정 기법이 있지만, 하나의 임계값으로는 세포영상에서 사용자가 원하는 세포들을 정확히 찾기에 어려움이 있다. 따라서 본 논문에서는 군집화(clustering)를 사용하여 1차로 객체(세포)와 배경을 분리하는 작업을 하고, 사용자가 분석하고자 하는 객체(세포)의 색상을 선택한 다음 해당 색상을 이용하여 영상을 이진화하는 방법을 사용한다. Fig. 4는 입력영상을 4개의 군집으로 군집화하고 사용자가 찾고자 하는 세포의 색을 선택한 후 이진화하는 과정을 나타낸다.

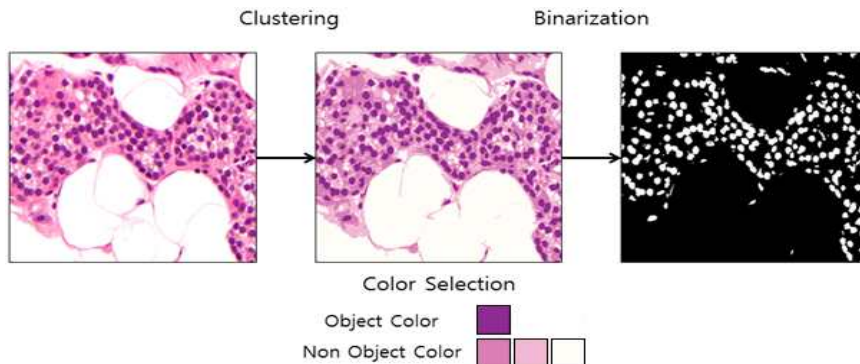


Fig. 4. Proposed image binarization procedure.

2.3 레이블링 영상 생성 및 모폴로지 연산

레이블링은 일반적으로 이진영상에서 수행되는 영역 구분 방법으로 동일 객체에 속한 모든 픽셀에 고유한 번호를 매기는 작업을 말한다[10]. 이진영상에서는 여러 객체(세포)를 하나로 보기 때문에 객체(세포)별 정보를 분석하는데 어려움이 있다. 따라서 레이블링을 수행하면 객체(세포)의 수를 알 수 있고, 각 객체(세포)를 레이블 번호로 구분할 수 있기 때문에 객체(세포)별 정보를 분석하는데 용이하다. 본 논문에서는 전통적 레이블링 기법을 사용하였다. 이 기법은 등가 테이블(equivalent table)을 만들면서 영상을 두 번 스캔함으로써 레이블링을 수행한다. 첫 번째 스캔에서는 레이블을 전파시키면서 등가 테이블

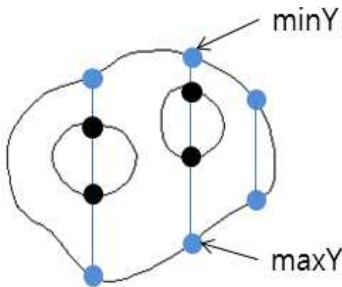


Fig. 8. minY and maxY in a column of a given cell.

을 만들고, 두 번째 스캔에서는 등가 테이블을 참조하여 각 픽셀에 고유의 레이블을 부여한다[10].

영상처리에서 모폴로지 연산은 영상을 형태학적인 관점에서 다루는 기법으로서 전처리 또는 후처리의 형태로 사용된다. 영상에서 잡음을 제거하거나 객체의 모양을 기술하는 용도 등으로 사용되며 가장 기본이 되는 연산은 침식(erosion)과 팽창(dilation)이다. 침식연산은 일반적으로 영상 내에서 객체 영역을 깎아내는 효과를 나타내며, 팽창연산은 객체 영역을 확장 시키는 결과를 만들어낸다. 침식연산과 팽창을 수식으로 정의하면 다음과 같다[10].

$$A \ominus B = \{x | (B)_x \subset A\} \tag{2}$$

$$A \oplus B = \{x | [(\hat{B})_x \cap A] \subseteq A\} \tag{3}$$

여기서 A 는 입력영상을 나타내고 B 는 모폴로지 연산의 결과를 조정하는 구성요소를 나타낸다. 그러므로 모폴로지 연산을 수행하기 전에는 구성요소의 모양을 미리 결정해주어야 한다[10]. 본 논문에서는 3×3 크기의 정사각형 형태의 구성요소를 사용하여

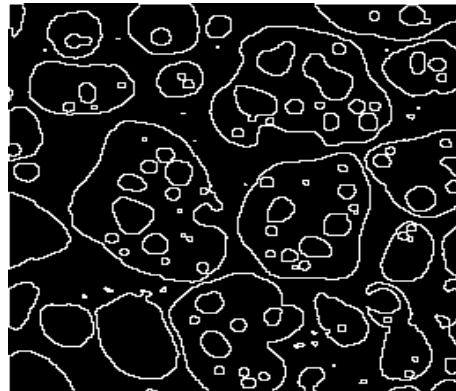


Fig. 5. Parallel contour line extraction result.

모폴로지 연산을 수행하였다.

Fig. 5는 전처리를 수행하여 추출된 외곽선 영상을 나타내는데, 입력영상의 레이블링 영상과 침식 연산을 수행한 레이블링 영상과의 차 연산을 통해 레이블링된 객체(세포)의 외곽선만을 추출한 결과이다. 여기서 많은 내부 외곽선이 발생한 것을 확인할 수 있다.

2.4 병렬 내부 외곽선 제거

본 논문에서는 외곽선의 좌표정보를 이용하여 내부 외곽선을 병렬로 제거하는 방법을 제안한다. 각 객체(세포)별로 내부 외곽선을 제거하기 위해서는 각 객체들 간에 외곽선의 좌표정보에 대하여 서로 영향을 미치지 않아야 한다. 따라서 전처리 과정에서 레이블링된 외곽선 영상을 생성하고, 각 kernel들은 같은 레이블을 가진 외곽선 좌표들만 비교하여 객체(세포)별 내부 외곽선 제거를 수행한다. CUDA에서 X, Y 좌표는 식(4)와 식(5)로 정의 된다. 식(4)와 식(5)에서 $blockIdx$ 는 각각 x, y 방향에 대한 블록 번호를 나타내고 $threadIdx$ 는 각 블록에 생성된 x, y 방향에 대한 스레드 번호를 나타낸다. $blockDim$ 은 블록에 생성된

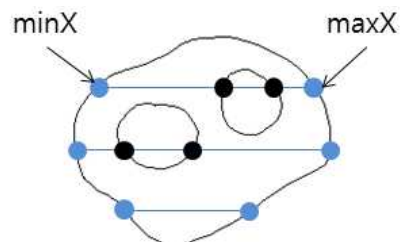


Fig. 6. minX and maxX in a line(row) of a given cell.

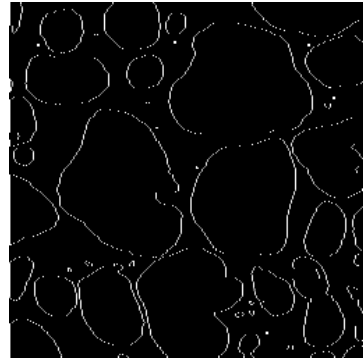
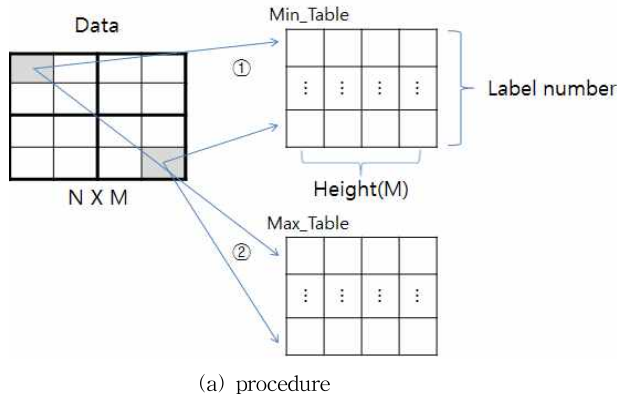


Fig. 7. Kernel_1 (extraction of minX and maxX).

한 행, 열에 존재하는 스프레드 개수를 의미한다[11].

$$X = blockIdx.x \times blockDim.x + threadIdx.x \quad (4)$$

$$Y = blockIdx.y \times blockDim.y + threadIdx.y \quad (5)$$

Kernel_1은 레이블링된 외곽선 영상에서 같은 레이블을 갖고 Y좌표가 같은 픽셀의 X값의 최소, 최대값을 탐색한다. Y좌표가 같다는 것은 2차원으로 표현된 영상에서 한 행을 의미한다. Fig. 6은 내부 외곽선이 존재하는 한 객체에서 Kernel_1을 수행하여 최소, 최대 X좌표를 탐색한 것을 보여준다. Fig. 6에서 최소, 최대 X좌표는 minX와 maxX(파란점)로 표시되어 있다.

탐색된 minX와 maxX는 각각 Min_Table, Max_Table에 갱신된다. Fig. 7은 최소, 최대 X좌표 정보가 Min_Table과 Max_Table에 병렬로 갱신되는 과정을 나타낸다. 여기서 Data는 입력 영상이며, 2x2

CUDA 블록으로 구성된 예이다.

Kernel_2는 레이블링(Lableing)된 외곽선 영상에서 같은 레이블을 갖고 X좌표가 같은 픽셀의 Y값의 최소, 최대값을 탐색한다. X좌표가 같다는 것은 2차원으로 표현된 영상에서 한 열을 의미한다. Fig. 8은 내부 외곽선이 존재하는 한 객체에서 Kernel_2를 수행하여 최소, 최대 Y좌표를 탐색한 것을 보여준다. Fig. 8에서 최소, 최대 Y좌표는 minY와 maxY(파란점)로 표시되어 있다.

탐색된 minY와 maxY는 각각 Min_Table, Max_Table에 갱신된다. Fig. 9는 최소, 최대 Y좌표 정보가 Min_Table과 Max_Table에 병렬로 갱신되는 과정을 나타낸다.

Kernel_3에서는 Kernel_1, 2에서 얻어진 최소 X, Y좌표와 최대 X, Y좌표를 이용하여 최종 외곽선을 추출하는 작업을 수행한다. 최소 X, Y좌표와 최대

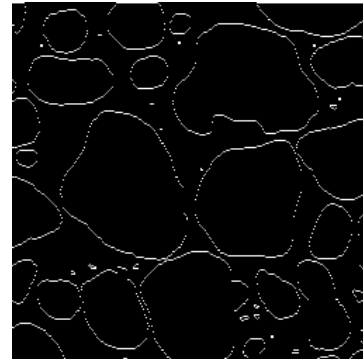
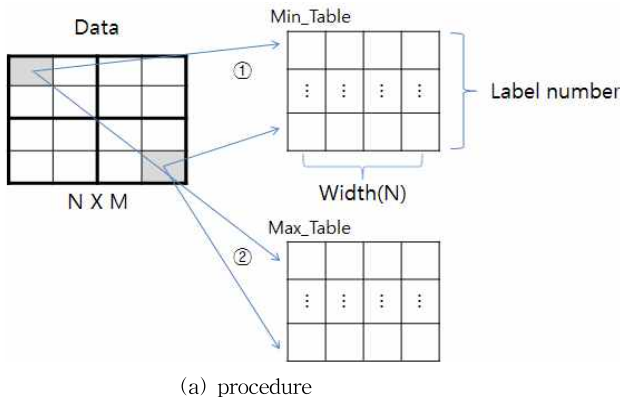


Fig. 9. Kernel_2 (Extraction of minY and maxY).

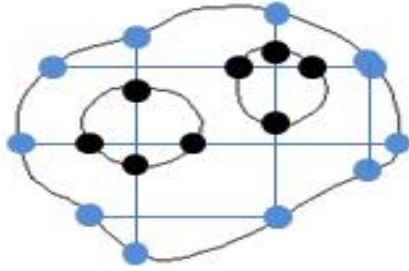


Fig. 10. Logical sum for min_X, min_Y, max_X and max_Y.

X, Y 좌표의 논리합은 객체(세포)의 최외곽선을 이루는 외곽점들의 좌표들을 의미하므로 내부 외곽선이 제거된 결과를 얻을 수 있다. Fig. 12는 최소 X, Y 좌표와 최대 X, Y 좌표에 대한 논리합 결과를 나타낸다.

탐색된 최소 X, Y 좌표와 최대 X, Y 좌표 정보는 각각 Min_Table_X, Min_Table_Y, Max_Table_X, Max_Table_Y에 저장되어 있는데, Fig. 11은 각 테이블에 저장된 좌표정보를 이용하여 병렬로 논리합을 수행하는 것을 나타낸다.

3. 실험

본 논문의 실험을 위해 Win7환경 하에서 순차처리 시스템과 CUDA를 이용한 병렬처리 시스템을 구현하였다. 순차처리 시스템은 기본적인 외곽선 추적 알고리즘을 사용하여 구성하였고, 병렬처리 시스템은 본 논문에서 제안한 내부 외곽선 제거 알고리즘이 포함된 모폴로지 연산을 이용한 외곽선 탐색 알고리즘을 사용하여 구성하였다. 두 시스템 모두 전처리

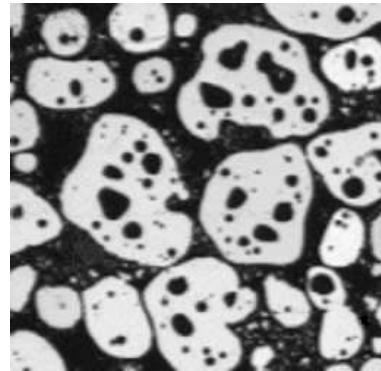
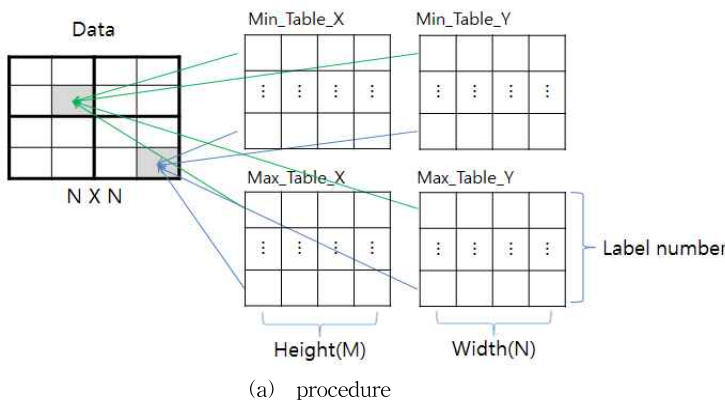


Fig. 12. Test image.

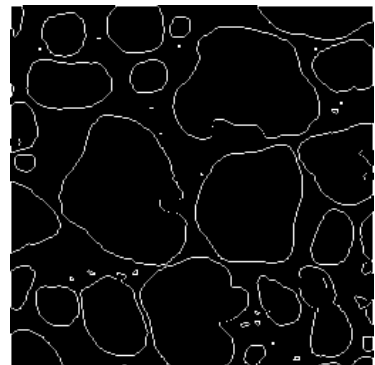
과정은 동일하며 레이블링은 병렬화에 어려움이 있어 순차처리 알고리즘으로 사용하였다. 따라서 전처리 부분에서 이진화 알고리즘과 외곽선 추적 알고리즘에 대한 성능만 비교 분석하였다. 성능 측정에는 256×256에서 4096×4096 크기의 영상을 사용하였고 (Fig. 12), 동일한 영상에 대하여 10번 이상 수행한 결과의 평균치를 계산하였다.

Table 1과 Fig. 13은 전처리 부분에 대한 성능을 비교한 결과이다. 입력영상의 크기가 커질수록 순차처리 알고리즘에 비하여 병렬처리 알고리즘이 더 빠르게 동작하는 것을 알 수 있고, 1024×1024이상의 영상에서는 10배 이상의 성능향상이 있음을 알 수 있다.

본 논문에서 제안하는 병렬 외곽선 추출 알고리즘의 경우, 잡음제거를 위한 모폴로지 연산 중 열림 연산을 수행한 시간과 외곽선 탐색 시간, 내부 외곽선 제거 시간을 합산하여 성능을 측정하였다. Table 2와



(a) procedure



(b) result image

Fig. 11. Kernel_3 (logical sum of X and Y).

Table 1. Binarization algorithm performance (Unit : ms)

image size	performance		
	serial(SD)	parallel(SD)	ratio
256×256	1.37(0.47)	0.41(0.02)	3.31
512×512	4.56(0.68)	0.65(0.05)	7.07
1024×1024	15.82(0.71)	1.38(0.06)	11.47
2048×2048	54.75(0.55)	4.32(0.29)	12.68
4096×4096	187.26(1.40)	14.25(0.42)	13.14

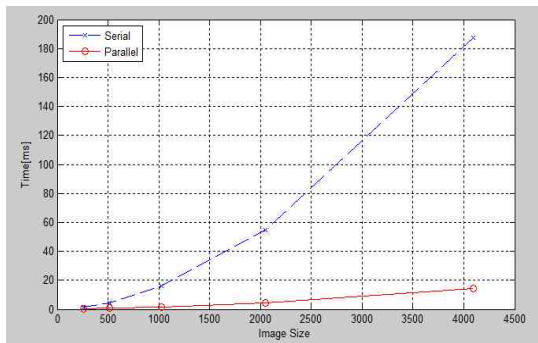


Fig. 13. Binarization algorithm performance.

Fig. 14는 외곽선 추출 알고리즘에 대한 성능을 비교한 결과이다. Table 2를 보면 순차처리 시스템이 병렬처리 시스템보다 성능이 빠른 것으로 나타나 있다. 이는 병렬처리 시스템의 경우 GPU로 영상 데이터를 전송하는 시간까지 포함되어 있는데, 각 알고리즘의 성능을 측정하기 위하여 독립적으로 동작하는 알고리즘들의 수행시간을 합산했기 때문이다. 다시 말해서 영상 데이터를 전송하는 시간이 여러 번 포함되어 있다.

그러나 외곽선 추출 알고리즘은 영상이 입력되고 전처리 과정을 거친 후 동작하기 때문에 실제로는 영상 데이터를 전송하는 시간은 무시해도 무방하다.

Table 2. Contour line extraction performance including data moving (unit : ms)

image size	performance		
	serial(SD)	parallel(SD)	ratio
256×256	2.55(0.46)	7.11(1.10)	0.36
512×512	4.53(0.06)	13.85(2.35)	0.33
1024×1024	10.09(0.17)	24.44(4.97)	0.41
2048×2048	24.80(0.36)	49.50(4.34)	0.50
4096×4096	68.70(1.20)	112.83(4.84)	0.61

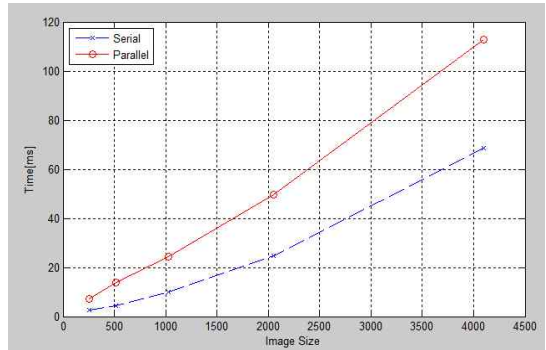


Fig. 14. Contour line extraction performance.

따라서 영상 데이터를 전송하는 시간을 제외한 성능 측정을 하였고 이는 Table 3과 Fig. 15에 나타내었다. 병렬처리 시스템이 순차처리 시스템에 비하여 4.1~7.6배의 성능 향상이 있음을 알 수 있다. 영상의 크기가 커질수록 성능이 향상되지 않는 이유는 잡음제거의 시간이 커졌기 때문이다. 또한 알고리즘들에 대한 최적화 및 기능 개선을 통해 좀 더 나은 성능을 보일 것으로 기대된다.

Table 3. Pure contour line extraction performance excluding data moving (unit : ms)

image size	performance		
	serial(SD)	parallel(SD)	ratio
256×256	2.55(0.46)	0.33(0.04)	7.69
512×512	4.53(0.06)	0.60(0.06)	7.60
1024×1024	10.09(0.17)	1.31(0.02)	7.73
2048×2048	24.80(0.36)	4.31(0.13)	5.75
4096×4096	68.70(1.20)	16.78(0.06)	4.10

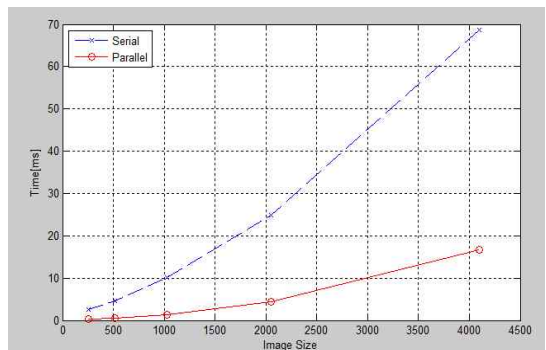


Fig. 15. Pure contour line extraction performance.

4. 결 론

본 논문에서는 병렬로 세포 영역에서 외곽선을 추출하는 알고리즘과 외곽선을 추출할 때, 생성되는 내부 외곽선을 병렬로 제거하는 방법을 제안하였다. 제안 알고리즘은 외곽선 정보를 저장하지 않고 영상만 획득하여 사용하기 때문에 외곽선 정보를 저장하는데 사용되는 메모리 공간을 줄일 수 있고, 기존의 외곽선 추적 알고리즘보다 4.1~7.6배 나은 성능을 보여주었다.

그러나 객체가 불록 도형이 아닌 경우 즉, 최외곽선에 오목한 부분이 존재할 경우 외곽선을 잃어버리는 문제점이 있다. 향후 과제로는 이런 문제점을 개선하여 어떤 임의의 객체에도 적용할 수 있는 알고리즘에 대한 연구가 필요할 것으로 생각된다.

REFERENCE

- [1] C.H. Cheng and T.D. Han, "Improved Simple Boundary Following Algorithm," *Journal of Korean Institute of Information Scientists and Engineers: Software and Applications*, Vol. 33, No. 4, pp. 427-439, 2006.
- [2] R.O. Duda and P.E. Hart, *Pattern Recognition and Scene Analysis*, John Wiley & Sons, New York, 1973.
- [3] E. Gose, R. Johnson, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall, New Jersey, 1996.
- [4] A. Mirante and N. Weingarten, "The Radial Sweep Algorithm for Constructing Triangular Irregular Networks," *Journal of IEEE Computer Graphics and Applications*, Vol. 2, No. 3 pp. 11-21, 1982.
- [5] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 1982.
- [6] M.N. Kim and J. H. Cho, "The Automatic Detection of Inner Boundary on EBCT Images for Airway," *Journal of Korea Multimedia Society*, Vol. 6, No. 6, pp. 991-999, 2003.
- [7] W.G. Jeon, T.Y. Kim, S.J. Kim, H.K. Choi, and K.G. Kim, "Lung Segmentation Considering Global and Local Properties in Chest X-ray Images," *Journal of Korea Multimedia Society*, Vol. 16, No. 7, pp. 829-840, 2013.
- [8] S.W. Choi and S.H. Yu, "Area Measurement of Organism Image using Super Sampling and Interpolation," *Journal of Korea Multimedia Society*, Vol. 17, No. 10, pp. 1150-1159, 2014.
- [9] H.S. Lee and H.Y. Kwon, "Parallel Cell Contour Line Extraction using CUDA," *Proceedings of the Spring Conference Korean Multimedia Society*, Vol. 18, No. 1, pp. 206-209, 2015.
- [10] R.C. Gonzalez and R.E. Woods, *Digital Image Processing-3/E*, Prentice Hall, New Jersey, 2007.
- [11] D.B. Kirk and W.W. Hwu, *Programming Massively Parallel Processors*, BJ Public, Goyang, 2010.



이 호 석

2014년 안양대학교 컴퓨터공학과
학사
2014년~현재 안양대학교 컴퓨터
공학과 석사과정
관심분야 : 영상처리



권 희 용

1983년 서울대학교 전자계산기공
학과 학사
1985년 서울대학교 전자계산기공
학과 석사
1993년 서울대학교 컴퓨터공학과
박사



유 숙 현

1999년 안양대학교 컴퓨터공학과
학사
2002년 안양대학교 컴퓨터공학과
석사
2011년 안양대학교 컴퓨터공학과
박사

1986년~1995년 한국통신 연구개발단 선임연구원
1995년~현재 안양대학교 컴퓨터공학과 교수
관심분야 : 패턴인식, 신경망, 영상처리, 병렬처리응용

2002년~2011년 안양대학교, 대림대학 출강
2012~현재 안양대학교 정보통신공학과 조교수
관심분야 : 패턴인식, 신경망, 영상처리, 병렬처리응용