

Slice 정보에 기반한 3차원 볼륨 데이터의 레이블링 알고리즘

(A Slice Information Based Labeling Algorithm for 3-D Volume Data)

최 익 환 [†] 최 현 주 ^{**} 이 병 일 ^{***} 최 흥 국 ^{****}
 (Ikhwan Choi) (Hyunju Choi) (Byeongil Lee) (Heungkook Choi)

요약 본 논문에서는 볼륨데이터에서의 레이블링(labeling)을 위한 알고리즘을 제안하고자한다. 3차원 볼륨은 2차원 슬라이스 데이터의 연속으로 보고 각 슬라이스의 레이블링 정보를 바탕으로 하는 SIL(Slice Information based Labeling)방법을 제안한다. 이는 기존의 알고리즘에 비해 효율적인 메모리 사용이 가능하고 분석하고자 하는 데이터의 특성에 맞는 2차원 레이블링과의 조합이 가능한 장점이 있다. 기존 알고리즘과 제안하는 방법을 3차원 세포영상에서 비교하여 보았으며, SIL을 2차원 레이블링 CCCL(Contour based Connected Component Labeling)과 함께 볼륨데이터에 적용하여 본 결과 기존의 알고리즘 보다 약 2배 빠른 성능을 보였다. 다양한 3차원 레이블링 방법 중 적용되는 영상에 따라 각기 다른 결과를 얻었지만, 3차원 세포영상의 분석에서는 SIL 방법이 우수하다는 결론을 얻었다.

키워드 : 볼륨데이터, SIL, CCCL, 3차원 레이블링

Abstract We propose a new 3 dimensional labeling method based on slice information for the volume data. This method is named SIL (Slice Information based Labeling). Compare to the conventional algorithms, it has advantages that the use of memory is efficient and it is possible to combine with a variety of 2 dimensional labeling algorithms for finding an appropriate labeling algorithm to its application.

In this study, we applied SIL to confocal microscopy images of cervix cancer cell and compared the results of labeling. According to the measurement, we found that the speed of SIL combined with CCCL (Contour based Connected Component Labeling) is almost 2 times higher than that of other methods. In conclusion, considering that the performance of labeling depends on a kind of image, we obtained that the proposed method provide better result for the confocal microscopy cell volume data.

Key words : Volume data, SIL, CCCL, 3D labeling

1. 서론

레이블링(Labeling)을 통해 객체(Object)를 분리하는 것은 영상분석에서 가장 중요한 과정 중의 하나이며 이는 객체에 고유한 번호를 부여함으로써 서로 떨어진 객체들을 구별해 내는 방법을 말한다. 물체분석, 영상분할

및 병합, 잡음제거, 인식 등 여러 분야에 활용되고 있다 [1,2,3]. 이러한 레이블링 기법은 예전부터 Rosenfeld와 Pfaltz 등에 의해 개발되어 많은 개선 및 연구 되어왔다 [4,5]. 일반적인 레이블링 알고리즘은 이진화시킨 영상에 대해 두 번의 좌측에서 우측으로 그리고 위에서 아래로 살펴보는 스캔과정을 통해 객체를 레이블링 한다. 첫 번째 스캔에서 현재 마스크의 중앙픽셀(pixel)에 임시 레이블을 부여한다. 임시 레이블은 이웃하는 픽셀의 레이블 값 중 최소값을 준다. 만약 처음 접하는 픽셀인 경우에는 새로운 레이블을 부여한다. 같은 객체 내에서 레이블이 서로 일치하지 않는 경우는 동일한 레이블이지만 다른 레이블로 처리된 것으로 이러한 다중레이블에 대한 처리는 동치레이블의 동치 레이블 집합에서 최소값을 찾아 같은 레이블로 처리하여 줌으로써 일치된 하나

[†] 비회원 : 엔솔테크 연구원
minibug@chollian.net

^{**} 학생회원 : 인제대학교 전산학과
hjchoi@mitl.inje.ac.kr

^{***} 비회원 : 에디트 대표
kleo@nm.snu.ac.kr

^{****} 비회원 : 인제대학교 컴퓨터공학부 교수
hkchoi@mitl.inje.ac.kr

논문접수 : 2004년 2월 12일

심사완료 : 2004년 4월 16일

의 고유한 레이블을 얻게 된다. 이 병합(Merge)과정이 끝나면 두 번째 스캔에서 임시 레이블에 대하여 우측에서 좌측으로 그리고 아래에서 위쪽으로 마스크를 이동하면서 중앙 픽셀의 우측 픽셀과 아래 픽셀을 살펴 보면서 해당하는 동치클래스의 고유한 레이블을 부여함으로써 완료하게 된다. 이 방법은 동치테이블의 효과적인 처리 및 구현을 위해 연결목록(Linked list)을 사용하고 동적인 테이블을 잡아야 하므로 구현이 다소 복잡할 뿐 아니라 처리 시간이 오래 걸리는 단점이 있다.

이를 개선하기 위한 방법으로 동치테이블 대신 boolean matrix를 사용하거나 동치테이블을 그래프 구조로 구현하여 depth-first search로 레이블을 구하는 방법이 제안되었다[6,7]. Stefano 등은 동치테이블을 구성하는 복잡함을 개선하여 1D class array와 frequency array를 두어 동치클래스에 해당하는 두 개의 레이블을 첫 번째 처리에서 바로 저장하는 방법을 사용하였다[8]. 하지만 이미지의 크기가 커지고 레이블의 개수가 늘어남에 따라 class array를 조작하는 시간이 많아져 수행 속도가 느려지는 경향이 있었다. 또한 효율적인 레이블 상충(Conflict)을 처리하기 위하여 방법으로 flood-fill 알고리즘을 이용한 한번의 스캔을 통하여 레이블 값을 조정하는 NFL(Non-recursive Flood-fill Labeling) 방법이 제안되었다[9].

최근에는 의료기기의 발달로 2차원 영상뿐만 아니라 CT, MRI나 LSCM(Laser Scanning Confocal Microscopy)같은 장비에서 얻어지는 3차원 볼륨데이터의 사용이 늘어나면서 볼륨데이터에서 객체의 분리에 대한 연구가 요구되고 있다. 일반적으로 3차원데이터의 처리시간은 2차원영상보다 상당히 크므로, 볼륨데이터에서 빠르고 효율적인 레이블링 알고리즘이 필요하다. 볼륨데이터의 객체에 레이블링하는 방법은 6방향, 18방향, 그리고 26 방향의 3차원 마스크를 사용하여 3차원데이터를 한번에 접근하여 처리하는 방법이 있는데 사용하는 클래스나 저장공간에 따라 메모리를 많이 필요로 하는 단점이 있다.

본 연구에서는 2차원 슬라이스(slice) 정보를 이용한 3차원 레이블링 방법을 제안한다. 각 슬라이스 별로 레이블링을 하여 얻어진 각 객체 정보들을 이용하면 각 객체가 이루고 있는 복셀(Voxel) 개개의 연결성을 따지지 않아도 상하 객체들 사이의 연결성을 쉽게 확인 할 수 있다. 성능 평가를 위하여 기존의 레이블링 방법에서 개선한 레이블링 방법을 몇 가지 영상에 대해 실험하고 그 결과를 비교해 보았다.

2장에서는 레이블링 알고리즘의 기존연구에 대하여 살펴보고 3장에서는 논문에서 제안한 CCCL(Contour based Connected Component Labeling)을 이용한

SIL(Slice Information based Labeling)에 대해서 알아 보고 4장에서는 앞의 알고리즘을 여러 영상에 테스트하여 알고리즘들을 비교 실험하였다. 그리고 5장에서는 향후 연구 방향 및 결론을 맺는다.

2. 기존 연구

볼륨데이터에서 레이블링의 기존연구로는 NFL을 볼륨데이터에서도 가능하도록 수정하는 방법과 GCCL(Greedy connected component labeling)등이 있다. NFL은 그래픽 툴에서 흔히 사용하는 색채음 방법인 flood-filling을 이용하는 방법이다[9]. 하지만 색채음 방법은 시스템 스택(Stack)을 사용한 재귀적인 방법으로 overhead를 발생 시킬 뿐만 아니라 stack memory dump fail의 가능성이 크다. 따라서 방문여부(isVisited)와 복귀위치(pReturnPos)를 변수로 가지는 LS(Label Stack) 클래스를 영상과 같은 크기를 할당하고 이미지의 각 픽셀을 방문하면서 방문여부를 변경하고 복귀 위치를 저장함으로써 시스템 스택 기능을 대신하게 된다.

알고리즘 1 NFL 알고리즘

```
label_number = 1
For (scan entire image)
  If !s.isVisited == false && current_pixel_value == target_value then
    Execute Non_recursive_flood_filling
    label_number ++
End for
```

알고리즘 2 Non_recursive_flood_filling 알고리즘

```
set target_value = pixel(x,y) value
current_position = pixel(x,y) position
end_position = pixel(x,y) position

Do
  current_position = visited
  pixel(current_position) = label_number
  If one of current_position's 4/8 connected position was not
  visited and pixel(x,y) value == target_value then
    return_position of correspond neighbor pixel = current_position
    current_position = correspond neighbor pixel's position
  Else
    current_position = current pixel's return_position
While(current_pixel != end_pixel)
```

구체적인 레이블링 방법은 이진화된 이미지에서 첫 번째 스캔을 통해 방문하지 않은 1인 픽셀에 대하여 non-recursive flood-filling을 수행한다. 여기에서는 현재 위치 및 종료 위치를 설정하고 방문 flag를 설정한 다음 현재 위치에 대하여 현재 레이블번호를 붙인다. 그 다음 주변 픽셀 중에서 방문하지 않은 1인 픽셀 n을 찾

는다. 이때 n이 존재 할 경우 픽셀 n의 복귀위치를 현재 픽셀 x의 위치로 설정해둔다. 그 다음 현재 픽셀의 위치를 다시 n의 위치로 설정하여 이를 반복함으로써 flood-filling이 구현된다. 이러한 과정은 알고리즘 2에 요약되어 있다. 만약 방문해야 할 주변 픽셀 n이 없을 경우, 현재 픽셀의 위치를 현재 픽셀의 복귀위치로 설정함으로써 이전 위치로의 이동이 이루어지게 된다. 만약 리턴 된 픽셀의 위치가 종료 위치와 일치되면 다음 레이블에 대한 레이블링을 계속하게 된다. 영역확장 방법으로 객체가 하나씩 레이블을 붙여 나간다. NFL알고리즘은 알고리즘 1에 요약되어 있다.

이 NFL알고리즘을 수정하면 쉽게 3D 레이블링이 가능한데, 스캔 방향에 z축을 추가하고 연결정보를 4/8방향에서 6/18/26방향으로 확장하고 현재위치정보와 복귀 위치 등의 변수들이 복셀 정보를 가지도록 수정하면 볼륨데이터에서의 레이블링이 가능해진다. 하지만 LS클래스의 경우 2바이트의 unsigned short형을 사용한 x, y, z 변수와 부울(bool)형 변수를 사용할 경우라도 LS클래스의 크기는 12바이트(VC++ sizeof()함수 사용)가 된다. 따라서 볼륨데이터의 크기만큼 LS를 선언해주게 되면 많은 메모리를 필요로 하는 단점이 있다.

GCCL(Greedy Connected Component Labeling)은 DL(Dynamic List)을 사용하며 한번의 스캔으로 레이블링을 수행한다[10]. 따라서 메모리 활용 측면에서 3D-NFL보다 좋다. GCCL은 이진화된 영상을 스캔 하면서 레이블이 없는 복셀을 찾으면 6/18/26 방향으로 연결된 모든 복셀이 레이블링이 될 때까지 레이블을 붙여나간다. 이렇게 레이블 붙인 복셀들은 각 레이블마다

알고리즘 3 GCCL 알고리즘

```

Read binary image T, and set t(i,j,k) = 1, if (i,j,k) ≠ background
while t(i,j,k) == 1 loop
    Generate a DL, denoted by dl and a temporary DL, denoted by tdl. Two counters, denoted by num1 and num2, represent the increase of dl and tdl. Add voxel (i,j,k) into tdl
    while tdl ≠ ∅ and voxel(l,m,n) ∈ tdl loop
        if t(l,m,n) == 1 or 2 then
            while all the 6/18/26-connected voxels of (l,m,n) and t(l±1,m±1,n±1) = 1 loop
                num1++
                Add voxel (l±1,m±1,n±1) into tdl
                Set t(l±1,m±1,n±1) = 2
            End loop
        Set t(l,m,n) = 3
        Add (l,m,n) into dl
        num2++
    End if
    Remove voxel (i,j,k) from the tdl
End loop
End loop
    
```

각각의 DL에 속하는 컴포넌트(Component)가 된다. 그러나 복셀이 중복 사용되거나 레이블이 재할당 되는 것을 막기 위해 GCCL에서는 4-value flag를 사용한다.

t(i,j,k)를 컴포넌트 복셀이라 하면, 0,1,2,3들의 값을 가질 수 있다. 만약 t(i,j,k)=0 이라면 배경에 해당하는 값이고 이 복셀의 값은 변하지 않는다. t(i,j,k)=1 이면 목표 복셀에 해당하는 값으로 레이블값이 할당 되지 않았고 또한 새로운 DL 시작이 될 수 있음을 의미한다. t(i,j,k)=2 이면 DL에 추가되는 새로운 노드(Node)를 의미한다. 하지만 새로운 레이블이 될 수 없다. t(i,j,k)=3 이면 이미 DL에 추가되었으며 새로운 노드가 될 수 없음을 의미한다. 이러한 알고리즘은 알고리즘 3에 요약되어 있다.

GCCL 알고리즘에서는 사용한 함수를 알아보면, 먼저 Add(x) 함수는 O(1)의 수행시간으로 DL에 복셀 x에 대한 노드를 생성하여 추가하는 함수이다. Remove(x) 함수는 DL의 첫 번째 포인터로 이동하여 복셀 x를 찾아서 존재하면 지우는 함수이다. 노드의 개수가 n일 경우 보통 O(n)의 수행시간을 가지지만, 이 알고리즘에서는 항상 root 노드를 가져와서 사용하고 지우게 되므로 O(1)의 수행시간을 가진다.

3. 3D 레이블링 알고리즘

볼륨데이터에서의 레이블링을 위해 본 논문에서 제안하는 SIL은 슬라이스들의 레이블링된 정보의 연결성을 검사하여 3D 레이블링 한다. 볼륨데이터는 그림 1과 같이 슬라이스 데이터의 집합과 같다[11]. 따라서 볼륨을 스캔 할 필요 없이 슬라이스에서 2D 레이블링된 정보를 추출하고 추출된 정보의 연관관계를 찾아줌으로써 3D 레이블링을 할 수 있다. 또한 슬라이스 별로 레이블링을 수행하므로 영상에 맞는 2D 레이블링 알고리즘을 선택하여 사용할 수 있다는 장점이 있다.

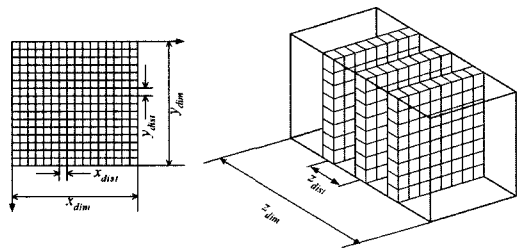


그림 1 볼륨데이터의 개요

본 논문에서 3D 레이블링을 위해 2D 레이블링 방법으로 CCCL을 사용한다. 이 CCCL은 Ishiyama등이 제안한 레이블링을 응용하는 방법으로 정수형 영역 정보

를 갖는 영상을 오직 한번 스캔 함으로써 모든 영역들의 정보를 추출할 수 있다[1,12].

기존의 제안된 방법은 객체의 영역의 경계를 추적하면서 가장 왼쪽의 픽셀에만 레이블을 붙여주고 그 픽셀의 레이블 값으로 객체의 레이블을 점차적으로 매겨나간다. 하지만 가장 왼쪽의 픽셀값만 레이블을 붙여주고 나가는 알고리즘은 다소 복잡해질 수 있다. 이는 watershed 등의 전처리를 거친 영상이 아니거나 형태가 복잡한 객체일 경우 외곽선을 전체를 추적해야 모든 왼쪽 픽셀에만 값을 할당 할 수 있으므로 객체의 외곽선을 추적해가는 알고리즘이 복잡해진다. 따라서 본 연구에서는 이 단계에서 레이블할당을 전체 외곽선에 적용함으로써 알고리즘을 간단히 하여 불필요한 연산을 줄이고자 하였으며 전처리 등에 의존하지 않게 하였다. 개선한 CCCL 레이블링 방법은 다음과 같다.

입력 영상과 동일한 크기의 버퍼를 할당하고 0으로 초기화한다. 각 객체들은 1부터 시작하여 차례대로 레이블이 증가하게 된다. 만약 현재 스캔하고 있는 위치의 레이블 값이 0인 경우에는 3가지로 나누어 처리된다. 먼저 가로 좌표가 0보다 크고 바로 왼쪽에 있는 점의 픽셀값과 현재 픽셀값이 동일한 경우, 버퍼의 바로 왼쪽의 픽셀과 동일한 레이블을 매긴다. 두 번째로는 가로 좌표와 세로 좌표가 모두 0이 아니고 현재 좌표에서 좌측 상단에 존재하는 픽셀값과 현재 픽셀의 값이 같지 않은 경우, 레이블을 1 증가시키고 현재 영역의 경계를 추적하면서 동일한 레이블을 적용한다. 세 번째에는 위의 두 경우를 만족하지 않는 경우는 객체 내의 홀로 인하여 같은 객체임에도 불구하고 새로운 객체로 인식될 수 있는 픽셀이므로 이를 피하기 위해 홀(hole) 추적을 시작한다. 이 때, 주위의 레이블을 검사하여 동일한 레이블을 매긴다. 스캔시 만약 현재 스캔하고 있는 위치에 이미 레이블이 매겨져 있다면 다음 픽셀로 이동한다. 이를 나타낸 것이 알고리즘 4이다.

각 슬라이스마다 레이블링이 끝나면 슬라이스별로 레이블 정보를 가지게 되며 이들 연결정보를 이용하여 3D 레이블링을 수행하게 된다. SIL 알고리즘은 다음과 같다. k번째 슬라이스의 객체 정보들과 k+1번째 슬라이스 객체 정보들을 비교하고 같은 3차원 객체에 해당하는지 검사하여 볼륨데이터의 레이블링을 구축한다. 이때 필요한 레이블링 정보로는 객체의 중심점, 영역크기, id가 필요하다. 비교하는 방법은 먼저 해당하는 k, k+1번째 객체들 영역의 교집합 여부를 판단한다. 없을 경우 비교 대상에서 제외된다. 겹쳐질 경우 중심점에 해당하는 픽셀들이 연결되었는가를 한번 더 비교한다. 이는 영역을 사각형형태로 비교하므로 실제 교집합 여부를 정확히 비교하고 비교 대상이 많을 경우 효율을 높이기 위하여

알고리즘 4 CCCL 알고리즘

```

label_number = 0
set buffer t, and t(x,y) = 0
while pixel(x,y) == 1 && t(x,y)==0 loop
  if t(x-1,y) has label && pixel(x,y) value == pixel(x-1,y) value
  then
    t(x,y) = t(x-1,y)
  else if pixel(x-1,y), pixel(x,y-1) and pixel(x,y) value are
  different each other then
    label_number ++
    Contour tracing and set t(contour) = label_number
  else Hole tracing and set t(hole) = label_number
  end if
end loop
    
```

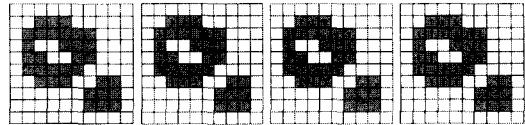


그림 2 CCCL의 진행과정

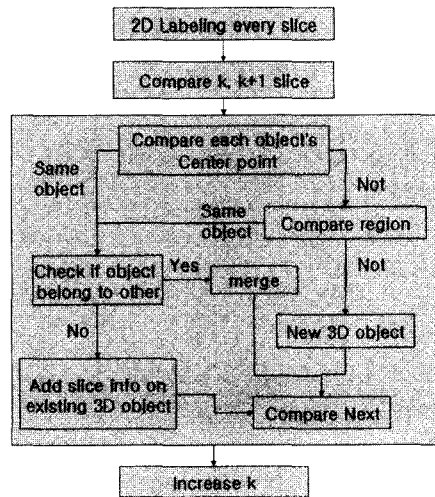


그림 3 SIL 알고리즘

다. 만약 도넛 모양처럼 중심점에 해당하는 픽셀이 비어 있거나 중심점에 해당하는 픽셀이 연결되지 않았을 경우에는 각 객체들의 픽셀들을 조사하여 연결 점을 형성하는가를 비교한다. 객체들이 연결 점을 가지지 않을 경우에는 각 객체들은 다른 객체들이며 k+1번째 객체의 경우 새로운 레이블이 된다. 연산량은 중심점을 먼저 확인하기 때문에 대부분의 경우 O(1)이다. 반면 각 픽셀들을 조사하는 경우에는 최악의 경우 O(k's object region ∩ k+1's object region)이 된다. 하지만 비교하다가 같은 객체 임이 밝혀지게 되면 종료하므로 실제 계산시간은 더 줄어든다. 그림 3은 SIL의 알고리즘 흐름

름도이다.

이 알고리즘은 k번째와 k+1번째를 비교함으로써 이루어지는데 이전에 새로운 레이블로 선언되었지만 그 이후에 같은 객체임이 밝혀지면 레이블 충돌이 발생할 수 있다. 주로 혈관영상 같이 나뭇가지 형태를 가진 객체일 경우에 많이 발생한다. 따라서 다른 객체에서 레이블 충돌이 발생할 때에는 병합을 수행하여 기존 레이블에 재적용한다. 이 병합 과정은 3D 레이블 정보를 DL로 저장하는 과정이다. 각 3D 객체는 각각의 DL에 추가되게 된다. 즉 다른 DL로 저장되었던 객체를 하나의 DL로 만들어주는 과정은 하나의 DL에 다른 DL 포인터를 연결 시켜줌으로써 쉽게 해결 할 수 있다.

4. 실험 결과

본 논문에서 제안한 알고리즘은 Visual C++ 6.0 MFC를 사용하여 Pentium-IV, 512MB메모리를 가진 시스템에서 구현하였으며, 서로 알고리즘을 비교하기 위하여 전처리 과정으로 이진화를 거친 데이터를 이용하였다.

표 1 이진화 시킨 실험영상

	Description	Size(pixel)
a	Bacterium	426x253
b	Circles	512x512
c	Brest cancer	512x512
d	Brest cancer	512x512
e	Uterine tissue	512x512
f	Brest cancer	640x480
g	Uterine cells	756x572
h	General cells	1280x960
i	General image	1600x1200

SIL에서 사용한 2D 레이블링의 성능을 평가하기 위해서 표 1의 데이터를 이용하여 비교하였으며 NFL방법과 GCCL 그리고 논문에서 제안하는 CCCL방법을 비교하였다. 그림 4의 결과에서는 전반적으로 0.5초 내외의 빠른 성능을 보여주지만 NFL은 (e)영상과 같이 특정영상에서 느려지는 현상이 있고 CCCL는 수행속도가 이미

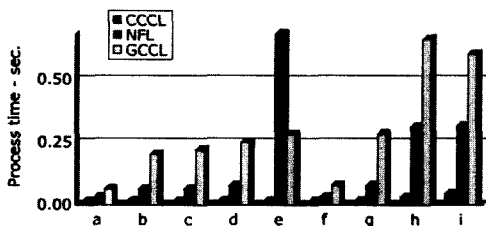


그림 4 2D 레이블링 결과

지 크기에만 의존하는 경향을 보인다. 하지만 (h, i)영상과 같이 크기가 커지더라도 제한하는 알고리즘의 수행속도가 상대적으로 매우 빠름을 알 수 있다.

3D 레이블링을 성능을 비교하기 위해서 표 2의 LSCM을 이용하여 획득한 자궁암 세포 영상 볼륨데이터 5set에 대해서 실험하였다.

표 2 실험 볼륨데이터

	Size(pixel)	Size(MB)
Set 1	512x512x37	9.28
Set 2	512x512x55	13.8
Set 3	512x512x117	29.3
Set 4	512x512x89	21.8
Set 5	512x512x50	12.5

3D 레이블링에서의 결과 비교는 SIL방법에 여러 레이블링 알고리즘: CCCL, NFL, GCCL을 사용한 방법들과 3D-NFL, GCCL방법, 5가지를 각각 비교하였다. 각 방법들의 수행 시간은 알고리즘의 수행과 연결된 객체들의 정보를 얻고, 일련의 메모리 할당 등의 연산처리가 포함된 총 처리시간이다. 그림 5에서 결과를 보면 set1의 경우 SIL(CCCL)이 0.937초, SIL(NFL) 1.657초, SIL(GCCL) 1.516초, 3D NFL 5.984초, GCCL 1.297초의 시간을 보이며, 제안한 SIL방법이 전체적으로 빠름을 알 수 있다. 3D-NFL의 경우에는 알고리즘 자체의 처리시간보다는 LS클래스 사용에서 오는 메모리할당에 많은 시간이 걸렸다. 특히 set3, set4의 볼륨데이터는 테스트환경에서 메모리 할당에 너무 많은 시간이 걸렸다. 반면 DL을 사용한 GCCL의 경우 NFL과 비교하여 볼륨데이터에서 성능이 좋았으며 GCCL이 좀더 효율적임을 알 수 있었다. 실험 데이터를 이용한 다섯 가지 종류의 테스트에서 본 논문에서 제안한 방법인 CCCL을 사용한 SIL이 좋은 성능을 보임을 알 수 있었다.

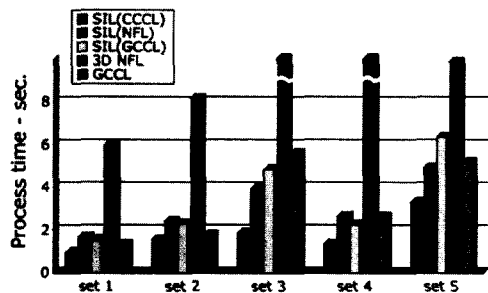


그림 5 3D 레이블링 결과

그림 6은 SIL에서 각각의 2D 레이블링과 SIL처리 루틴이 걸리는 시간을 측정된 것이다. 즉, 그림 6의 set1

은 0.937초가 걸리는데 이는 set1 결과에서 보듯이 2D 레이블링인 CCCL을 처리하는데 0.36초 SIL 처리 루틴으로 0.45초가 걸렸음을 말한다. 여기에서 남아있는 0.127초는 메모리 할당 등의 기타 연산이 걸리는 시간이다. SIL의 내부 처리 시간을 살펴보면 Set5 Result와 같이 영상이 복잡해 질수록 SIL의 시간 소모가 더 커지는 것을 알 수 있다.

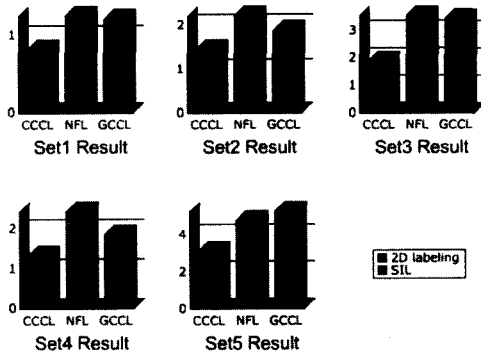


그림 6 각 영상의 결과비교

4. 결론

본 연구에서는 기존의 블록데이터에서의 레이블링에 대해 다른 접근방법으로 슬라이스 기반한(Slice-based) 3D 레이블링 알고리즘을 제안하였으며, 다른 방법들에 비해 좋은 결과를 얻을 수 있었다. 3D-NFL의 경우 알고리즘 자체만을 보면 빠른 시간을 보여주나, LS클래스 등의 선언으로 동적 메모리 할당을 사용함으로 인해 특히 할당을 위한 프로세스 타임이 소모가 크며, 메모리 할당량도 커서 효율적인 면에서 좋지 못하기 때문에 이를 위한 개선이 요구된다. GCCL의 경우 전반적으로 좋은 성능을 보이지만 메모리 할당부분에 있어서 개선이 이루어진다면 좀 더 빠른 시간을 보일 수 있다. 반면 SIL의 경우 한 장의 슬라이스 크기만큼만 메모리 할당이 이루어지고 재사용이 되므로 메모리 소비가 작고, 메모리 할당으로 인한 프로세스 타임이 없어진다. 실험결과를 봤을 때 전반적으로 CCCL을 이용한 SIL방법이 좋은 성능을 보일 뿐만 아니라 시스템적으로도 효율적인 알고리즘이라 할 수 있다.

향후 연구 방향으로는 좀 더 효율적인 슬라이스 레이블링 알고리즘의 개발과 슬라이스 객체 비교에서 발생할 수 있는 불필요한 연산의 최적화 방법의 개발, 그리고 효율적인 메모리 사용과 프로세서 관리 등을 추진할 것이다.

참고 문헌

- [1] 황선규, 권영진, 김희율, "영상 분할을 위한 효과적인 영역 병합 방법", 제 15 회 신호처리 합동학술대회 논문집, pp.73, 2002.
- [2] 김태우, "MR영상의 3차원 가시화 및 분석을 위한 뇌 영역의 자동 분할", 정보처리논문지, 제7권, 제2호, pp.542-551, 2000.
- [3] Aronsson M. and Borgenfors G., "2D segmentation and labeling of clustered ring shaped objects," SCIA, pp. 272-279. 2001.
- [4] Rosenfeld A. and Pfalts J., "Sequential operations in digital picture processing," J. ACM Vol.13, No. 4, pp. 471-494. 1966.
- [5] Rosenfeld A. and Kak A. C., Digital Picture Processing, Academic Press, Vol.2, pp.241-242, 1982.
- [6] Gonzales R. and Woods R., Digital Image Processing, 2nd Ed., Addison-Wesley, New Jersey, pp.42-45, 2002.
- [7] Haralick R. and Shapiro L., Computer and Robot Vision Volume, Academic Press, Vol.2, pp.241-242, 1982.
- [8] Stefano L. D. and Bulgarelli A., "A simple and efficient connected component labeling algorithm," ICIAP '99, 10th International Conference on Image Analysis and Processing, September, pp. 27-29, 1999.
- [9] 김도현, 강동구, 차의영, "비재귀 Flood-Fill 알고리즘을 이용한 적응적 이미지 레이블링 알고리즘", 한국정보처리학회 논문지, 제9-B권, 제3호, pp.337-342, 2002.
- [10] Hu J., Farin G., II M. H., Massia S. P., Nielson G. and Razdan A., "Statistical 3D segmentation with greedy connected component labeling refinement," <http://prism.asu.edu/>, 2003.
- [11] Razan A., Patel K., Farin G. and Capco D. G. "Visualization of Multicolor LCM data set," Computers and Graphics, Vol.25, No.3, pp.371-382, 2001.
- [12] Ishiyama Y., Funaoka C, Kubo F., Takahashi H. and Tomita F., "Labeling board based on boundary tracking," 11th ICPR, pp.34-38, 1992.



최 익 환

2002년 인제대학교 컴퓨터공학부 학사
2004년 인제대학교 전산학과 석사. 2004년 현재 엔셀테크 연구원. 관심분야는 영상처리 및 분석, 컴퓨터 그래픽스



최 현 주

1995년 인제대학교 전산학과 학사. 2001년 인제대학교 전산학과 석사. 2004년 현재 인제대학교 전산학과 박사과정. 관심분야는 영상 처리 및 분석, 컴퓨터 그래픽스



이 병 일

1997년 인제대학교 전산학과 학사. 1999년 인제대학교 전산학과 석사. 2004년 인제대학교 전산학과 박사. 2000년~현재 메디트 대표, 고신대학교 겸임교수. 2004년 현재 서울대학교 의학연구원 방사선 의학연구소 선임연구원. 관심분야는 영상 처리 및 분석, 컴퓨터 그래픽스, 영상 검색



최 흥 국

1988년 Linkoeeping Univ. BS. 1990년 Linkoeeping Univ. MS. 1996년 Uppsala Univ. Ph.D. 1997년~현재 인제대학교 컴퓨터공학부 부교수. 한국멀티미디어학회 및 한국신호처리학회 국제이사. 관심분야는 멀티미디어, 컴퓨터 그래픽스, 영상처리 및 분석